Loan Approval

Trillyon Earl

2025-03-01

```r
library(MASS)
library(car)
library(glmnet)
library(nnet)
library(leaps)
library(dplyr)
library(tidyverse)
library(Sleuth3)
library(logistf)
library(brglm2)
```

Abstract

The data set chosen for this report was Loan Approval Classification. There are 14 predictor variables and 45000 observations. These predictor variables are:

Objective

The objective of this analysis is to accurately predict whether or not a person given some information about them would be approved for a loan. A value of 0 indicates rejecton and a values of 1 indicates acceptance for a loan in loan status.

Predictors	Type
Age	Float
Gender	Categorical
Education	Categorical
Income	Float
Employment Years	Integer
Home Ownership	Categorical
Loan Amount	Float
Loan Intent	Categorical
Loan Interest Rate	Float
Loan Percent Income	Float
Credit History Length	Float
Credit Score	Integer
Previous Loans	Categorical
Loan Status	Integer

Data Clean Up

```
loan <- read.csv("./loan_data.csv")
loan <- as_tibble(loan)
loan <- loan |> mutate(
  person_gender = factor(person_gender,
    ordered = FALSE, levels = c("female", "male")
  ),
  person_education = factor(person_education,
    ordered = TRUE,
    levels = c("High School", "Associate", "Bachelor", "Master", "Doctorate")
  ),
  person_home_ownership = factor(person_home_ownership,
    ordered = FALSE,
    c("RENT", "OWN", "MORTGAGE", "OTHER")
  ),
  loan_intent = factor(loan_intent,
    ordered = FALSE,
    c(
      "PERSONAL", "MEDICAL", "EDUCATION",
      "VENTURE", "HOMEIMPROVEMENT", "DEBTCONSOLIDATION"
    )
  ),
  previous_loan_defaults_on_file = factor(previous_loan_defaults_on_file,
    ordered = FALSE, c("Yes", "No")
  ),
  person_income = scale(person_income),
  loan_amnt = scale(loan_amnt),
)
set.seed(123)
n <- nrow(loan) # 45 000
idx <- sample.int(n, size = 0.8 * n) # random 80 %
train <- loan[idx, ] # 36 000 rows
test <- loan[-idx, ] # 9 000 rows
test$loan_status <- as.numeric(as.character(test$loan_status))
train$loan_status <- as.numeric(as.character(train$loan_status))

predictors_numeric <- train |>
  select(where(is.numeric)) |>
  select(-loan_status)
```

```
test_model <- function(model, testdat, cutoff = 0.5) {
  pred_prob <- predict(model, newdata = testdat, type = "response")
  pred_class <- ifelse(pred_prob > cutoff, 1, 0)
  data.frame(
    MSPE = mean((testdat$loan_status - predict(model, testdat, type = "response"))^2),
    Error = 1 - mean(pred_class == testdat$loan_status)
  )
}
```

Methods

Full Ordinary Logistic Model

This is the fitted model of all predictor variables that were left after data-clearning.

```r
full_model <- glm(loan_status ~ .,
  data = train,
  family = binomial(link = "logit")
)
summary(full_model)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.176e+01  1.146e+02  -0.190   0.8494
## person_age                       3.042e-02  1.223e-02   2.488   0.0128 *
## person_gendermale                2.433e-02  3.970e-02   0.613   0.5399
## person_education.L              -3.974e-02  1.076e-01  -0.369   0.7119
## person_education.Q              -2.797e-02  9.242e-02  -0.303   0.7622
## person_education.C              -6.312e-02  6.583e-02  -0.959   0.3377
## person_education^4              -5.586e-02  4.395e-02  -1.271   0.2037
## person_income                    3.546e-02  1.747e-02   2.029   0.0424 *
## person_emp_exp                  -2.269e-02  1.089e-02  -2.083   0.0372 *
## person_home_ownershipOWN        -2.131e+00  1.072e-01 -19.879  < 2e-16 ***
## person_home_ownershipMORTGAGE   -7.324e-01  4.494e-02 -16.298  < 2e-16 ***
## person_home_ownershipOTHER      -3.889e-01  3.569e-01  -1.090   0.2759
## loan_amnt                       -6.385e-01  2.781e-02 -22.961  < 2e-16 ***
## loan_intentMEDICAL               4.800e-01  6.519e-02   7.363 1.80e-13 ***
## loan_intentEDUCATION            -1.287e-01  6.730e-02  -1.912   0.0559 .
## loan_intentVENTURE              -4.740e-01  7.254e-02  -6.535 6.36e-11 ***
## loan_intentHOMEIMPROVEMENT       7.754e-01  7.563e-02  10.252  < 2e-16 ***
## loan_intentDEBTCONSOLIDATION     7.944e-01  6.737e-02  11.790  < 2e-16 ***
## loan_int_rate                    3.375e-01  7.379e-03  45.741  < 2e-16 ***
## loan_percent_income              1.589e+01  3.441e-01  46.179  < 2e-16 ***
## cb_person_cred_hist_length      -1.138e-02  1.010e-02  -1.126   0.2601
## credit_score                    -9.074e-03  4.587e-04 -19.781  < 2e-16 ***
## previous_loan_defaults_on_fileNo  2.038e+01  1.146e+02   0.178   0.8588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38136  on 35999  degrees of freedom
## Residual deviance: 15852  on 35977  degrees of freedom
## AIC: 15898
##
## Number of Fisher Scoring iterations: 19
```

```r
test_model(full_model, test)
```

```
##         MSPE     Error
```

```
## 1 0.07304491 0.1061111
```

```r
# Statistics
AIC(full_model)
```

```
## [1] 15898.15
```

```r
BIC(full_model)
```

```
## [1] 16093.45
```

Null Ordinary Logistic Model

This is the model of only the intercept.

```r
null_model <- glm(
  loan_status ~ 1,
  data   = train,
  family = binomial(link = "logit")
)

test_model(null_model, test)
```

```
##        MSPE     Error
## 1 0.1729012 0.2223333
```

```r
anova(null_model, full_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: loan_status ~ 1
## Model 2: loan_status ~ person_age + person_gender + person_education +
##     person_income + person_emp_exp + person_home_ownership +
##     loan_amnt + loan_intent + loan_int_rate + loan_percent_income +
##     cb_person_cred_hist_length + credit_score + previous_loan_defaults_on_file
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1     35999      38136
## 2     35977      15852 22    22284 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(null_model)
```

```
## [1] 38138.34
```

```r
BIC(null_model)
```

```
## [1] 38146.83
```

Null Model vs. Full Model

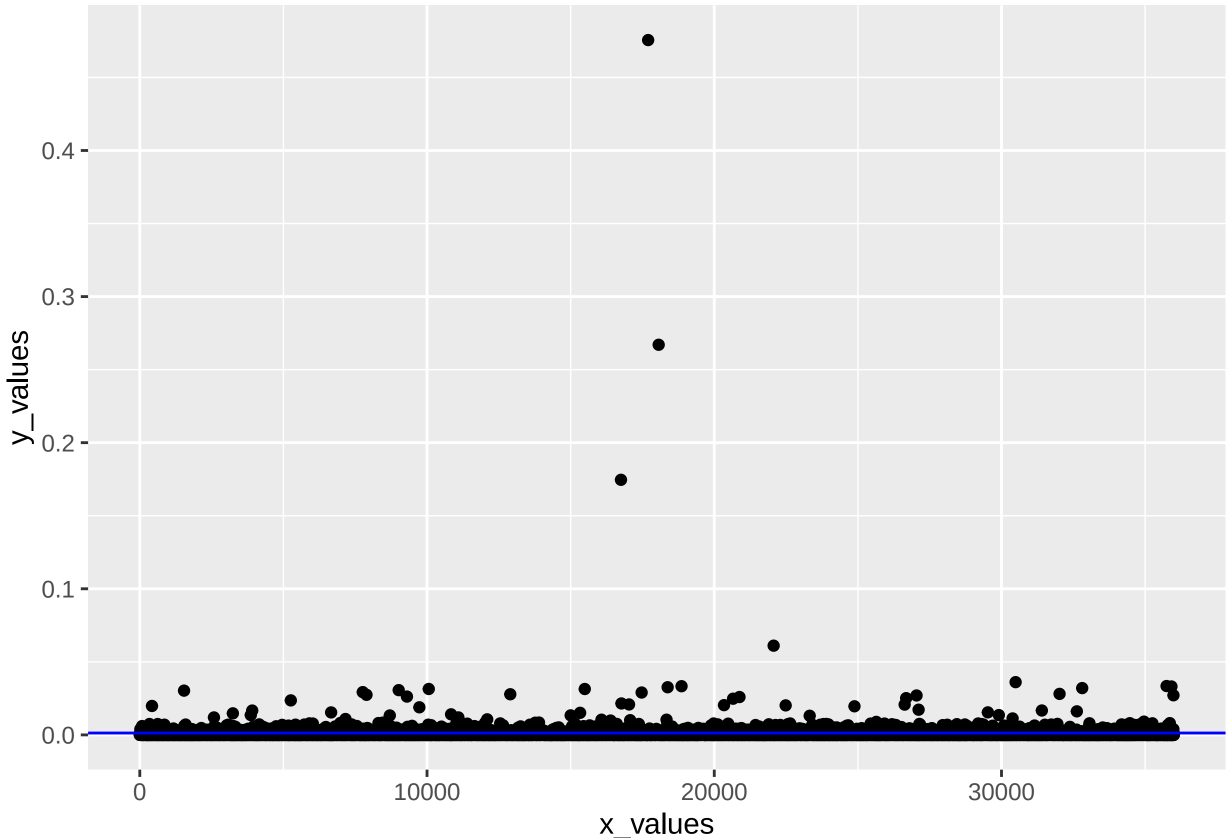
We can conclude from the statistics below that atleast one predictor variable is needed in our model as the we have gotten a very low p-value and the difference in AIC/BIC shows a very clear sign that we should not use the null model.

Problematic Sample Points

Leverage

We recieved around 5954 out of the 36000 oberservations that are considered high leverage.

```
lev <- hatvalues(full_model)
cutoff <- (2 * length(coef(full_model))) / nrow(train)
lev_data <- data.frame(
  x_values = seq_along(lev),
  y_values = lev
)
ggplot(aes(x = x_values, y = y_values), data = lev_data) +
  geom_point() +
  geom_hline(yintercept = cutoff, col = "blue")
```




```
high_lev <- which(lev > cutoff)
head(high_lev)
```

```
## 19 21 22 36 39 43
## 19 21 22 36 39 43
```

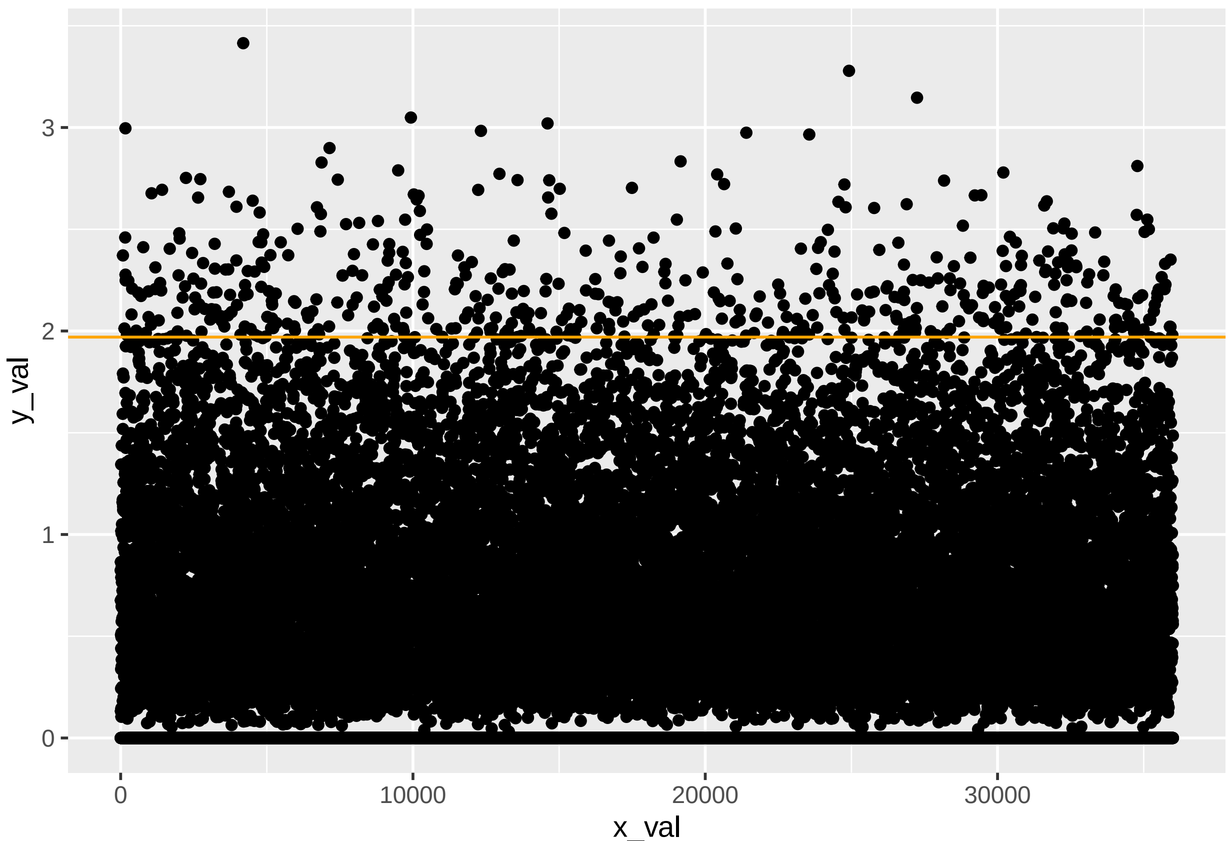
```
length(high_lev)
```

```
## [1] 5954
```

Outliers

We found 504 outliers within this data.

```
rstd <- rstudent(full_model)
outliers <- which(abs(rstd) >= 1.97)
ggplot(aes(x = x_val, y = y_val), data = data.frame(x_val = seq_along(rstd), y_val = abs(rstd))) +
  geom_point() +
  geom_hline(yintercept = 1.97, col = "orange")
```




```r
head(outliers)
```

```
##  77 129 155 163 165 170
##  77 129 155 163 165 170
```

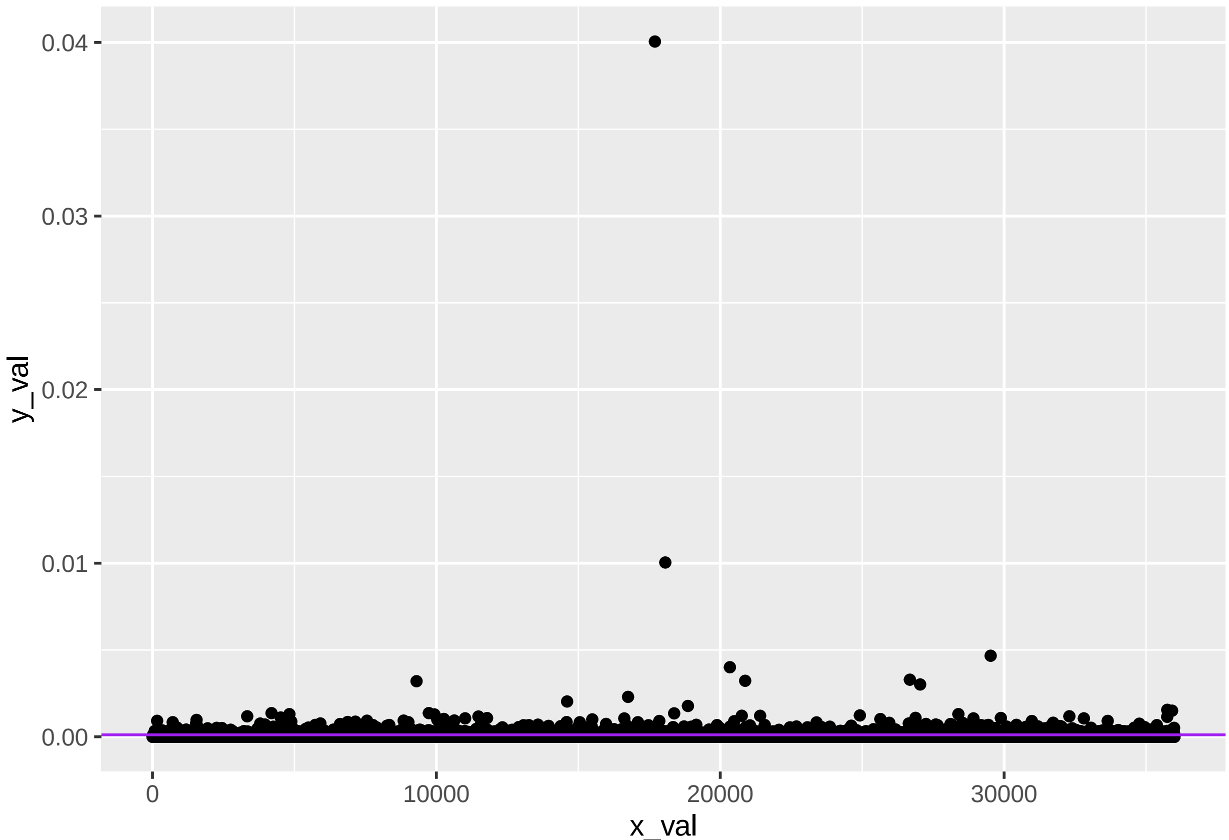
```r
length(outliers)
```

```
## [1] 504
```

Cook's Distance & Influential Points

Since 1 as the cutoff is too strict for this large dataset we adjusted the cutoff to be $\frac{4}{n}$ as used in a lot studies.

```r
cd <- cooks.distance(full_model)
influential <- which(cd > 4 / nrow(train))
influential <- unique(unlist(influential))
ggplot(aes(x = x_val, y = y_val), data = data.frame(x_val = seq_along(cd), y_val = cd)) +
  geom_point() +
  geom_hline(yintercept = 4 / nrow(train), col = "purple")
```




```
length(influential)
```

```
## [1] 2478
```

Problematic Points

We remove outliers that are influential points. From the R code below we can see that all of our outliers are influential points thust we should remove each of them.

```
sum(outliers %in% influential)
```

```
## [1] 504
```

```
train_no_outlier <- train[-outliers, ]

no_outlier_model <- glm(loan_status ~ .,
  data = train_no_outlier,
  family = binomial(link = "logit")
)

summary(no_outlier_model)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = binomial(link = "logit"),
##     data = train_no_outlier)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                       -2.360e+01  1.082e+02  -0.218 0.827445
## person_age                         5.779e-02  1.369e-02   4.223 2.41e-05 ***
## person_gendermale                  5.046e-02  4.432e-02   1.139 0.254881
## person_education.L                -9.888e-02  1.223e-01  -0.809 0.418667
## person_education.Q                -8.259e-02  1.050e-01  -0.787 0.431521
## person_education.C                -1.134e-01  7.441e-02  -1.524 0.127625
## person_education^4                -7.268e-02  4.925e-02  -1.476 0.140044
## person_income                      2.754e-02  2.656e-02   1.037 0.299797
## person_emp_exp                    -4.280e-02  1.219e-02  -3.511 0.000446 ***
## person_home_ownershipOWN          -2.782e+00  1.254e-01 -22.184  < 2e-16 ***
## person_home_ownershipMORTGAGE     -9.222e-01  5.077e-02 -18.165  < 2e-16 ***
## person_home_ownershipOTHER        -6.682e-01  4.015e-01  -1.664 0.096100 .
## loan_amnt                         -8.570e-01  3.327e-02 -25.755  < 2e-16 ***
## loan_intentMEDICAL                 6.798e-01  7.321e-02   9.286  < 2e-16 ***
## loan_intentEDUCATION              -1.496e-01  7.572e-02  -1.976 0.048191 *
## loan_intentVENTURE                -4.673e-01  8.143e-02  -5.739 9.53e-09 ***
## loan_intentHOMEIMPROVEMENT         1.104e+00  8.503e-02  12.985  < 2e-16 ***
## loan_intentDEBTCONSOLIDATION       1.082e+00  7.586e-02  14.270  < 2e-16 ***
## loan_int_rate                      4.581e-01  9.112e-03  50.270  < 2e-16 ***
## loan_percent_income                2.131e+01  4.370e-01  48.773  < 2e-16 ***
## cb_person_cred_hist_length        -2.553e-02  1.130e-02  -2.259 0.023909 *
## credit_score                      -1.180e-02  5.201e-04 -22.679  < 2e-16 ***
## previous_loan_defaults_on_fileNo  2.096e+01  1.082e+02   0.194 0.846446
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36984  on 35495  degrees of freedom
## Residual deviance: 12964  on 35473  degrees of freedom
## AIC: 13010
##
## Number of Fisher Scoring iterations: 19
```

```r
test_model(no_outlier_model, test)
```

```
##         MSPE     Error
## 1 0.07450024 0.1063333
```

```r
AIC(no_outlier_model)
```

```
## [1] 13010.07
```

```r
BIC(no_outlier_model)
```

```
## [1] 13205.05
```

Perfect and Quasi Complete Seperation

At this stage we have run into a problem. We encountered the warning message "glm.fit: fitted probabilities numerically 0 or 1 occurred". After researching this we came across that this is because we have Quasi-Complete Seperation. This means that one or more of our predictor varibles almost perfectly predicts our outcome variable.previous_loan_defaults_on_fileNo 2 From anaylzing the summary of the full model we can spot multiple problem covariates. This is a problem because it can lead to overfitting, and it pushes the maximum likelihood and standard error towards infinity.

Number of Previous Loan Defaults On File We can see that this covariate's estimate has a large magnitude around 20. This could be because of perfect and quasi complete seperation or multicollinearity.

Combatting Perfect and Quesi-Complete Seperation

From our research a useful way to combat Quesi-Complete Seperation is to merge or drop factor levels with too few observations (< 0.5% of data size) or drop levels that are heavily sided towards one outcome. The reason for this is that if these factor levels with few observations heavily lean towards one side of the binary outcome then when building a model it will see this almost perfect predictor for the outcome variable and thus give us an esitmate with a very large magnitude for that factor level. Factor levels with too few observations also may have a wide confidence interval. Thus we collpase these factor levels into a factor level that accounts for a larger group of observations. Below is a count of how many times each factor level occurs in the dataset of 36000 observations.

```r
table(train_no_outlier$loan_intent)
```

```
##
##          PERSONAL           MEDICAL         EDUCATION           VENTURE   HOMEIMPROVEMENT DEBTCONSOLI
##              5918              6751              7179              6180              3793
```

```r
xtabs(~ loan_status + loan_intent, data = train_no_outlier)
```

```
##            loan_intent
## loan_status PERSONAL MEDICAL EDUCATION VENTURE HOMEIMPROVEMENT DEBTCONSOLIDATION
##           0     4835    4889      6041    5316            2806              3965
##           1     1083    1862      1138     864             987              1710
```

```r
table(train_no_outlier$person_education)
```

```
##
## High School   Associate    Bachelor      Master   Doctorate
##        9468        9500       10573        5473         482
```

```r
xtabs(~ loan_status + person_education, data = train_no_outlier)
```

```
##            person_education
## loan_status High School Associate Bachelor Master Doctorate
##           0        7404      7452     8301   4316       379
##           1        2064      2048     2272   1157       103
```

```r
table(train_no_outlier$person_home_ownership)
```

```
##
##     RENT      OWN MORTGAGE    OTHER
##    18551     2329    14525       91
```

```r
xtabs(~ loan_status + person_home_ownership, data = train_no_outlier)
```

```
##            person_home_ownership
## loan_status  RENT   OWN MORTGAGE OTHER
##           0 12574  2172    13045    61
##           1  5977   157     1480    30
```

```r
table(train_no_outlier$previous_loan_defaults_on_file)
```

```
##
##   Yes    No
## 18259 17237
```

```r
xtabs(~ loan_status + previous_loan_defaults_on_file, data = train_no_outlier)
```

```
##            previous_loan_defaults_on_file
## loan_status   Yes    No
##           0 18259  9593
##           1     0  7644
```

We also see that we actually have complete seperation. Every observation that has had a previous loan default has been rejected for a loan. We dont want to just remove this level because it still contains very important information thus we dedcided to combat this by using firth logistic regression as recommended in the source provided.

Firth's Bias Reduced Logistic Regression

Firth logistic regression augments the ordinary log-likelihood with the **Firth penalty**

$$L_{\text{Firth}} = L_{\text{Logit}} \; - \; \frac{1}{2}\,\text{Penalty}$$

where the penalty term is made to prevent the log-likelihood from becoming infinite in cases of seperation.

Formula

$$\frac{1}{2}\,\log\bigl|I(\boldsymbol{\beta})\bigr|,$$

where $I(\boldsymbol{\beta})$ is the observed Fisher–information matrix.

```r
firth_model <- glm(
  formula = loan_status ~ ., data = train_no_outlier, family = "binomial",
  na.action = na.fail, method = "brglmFit"
)

summary(firth_model)
```

```
##
## Call:
## glm(formula = loan_status ~ ., family = "binomial", data = train_no_outlier,
##     na.action = na.fail, method = "brglmFit")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.24573  -0.12242  -0.00253  -0.00048   2.27273
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.393e+01  1.507e+00  -9.239  < 2e-16 ***
## person_age                      5.787e-02  1.367e-02   4.234 2.29e-05 ***
## person_gendermale               5.040e-02  4.428e-02   1.138 0.255039
## person_education.L             -9.829e-02  1.221e-01  -0.805 0.420892
## person_education.Q             -8.194e-02  1.049e-01  -0.781 0.434639
## person_education.C             -1.128e-01  7.433e-02  -1.518 0.128982
## person_education^4             -7.257e-02  4.920e-02  -1.475 0.140273
## person_income                   3.807e-02  2.063e-02   1.845 0.064986 .
## person_emp_exp                 -4.261e-02  1.218e-02  -3.499 0.000466 ***
## person_home_ownershipOWN       -2.776e+00  1.252e-01 -22.168  < 2e-16 ***
## person_home_ownershipMORTGAGE  -9.215e-01  5.070e-02 -18.174  < 2e-16 ***
```

```
## person_home_ownershipOTHER       -6.760e-01  4.009e-01  -1.686 0.091768 .
## loan_amnt                        -8.603e-01  3.224e-02 -26.684  < 2e-16 ***
## loan_intentMEDICAL                6.785e-01  7.314e-02   9.277  < 2e-16 ***
## loan_intentEDUCATION             -1.495e-01  7.565e-02  -1.977 0.048088 *
## loan_intentVENTURE               -4.665e-01  8.134e-02  -5.735 9.73e-09 ***
## loan_intentHOMEIMPROVEMENT        1.102e+00  8.495e-02  12.969  < 2e-16 ***
## loan_intentDEBTCONSOLIDATION      1.080e+00  7.578e-02  14.252  < 2e-16 ***
## loan_int_rate                     4.572e-01  9.096e-03  50.266  < 2e-16 ***
## loan_percent_income               2.132e+01  4.270e-01  49.936  < 2e-16 ***
## cb_person_cred_hist_length       -2.591e-02  1.127e-02  -2.299 0.021495 *
## credit_score                     -1.177e-02  5.194e-04 -22.661  < 2e-16 ***
## previous_loan_defaults_on_fileNo  1.129e+01  1.439e+00   7.844 4.37e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36984  on 35495  degrees of freedom
## Residual deviance: 12965  on 35473  degrees of freedom
## AIC:  13011
##
## Type of estimator: AS_mixed (mixed bias-reducing adjusted score equations)
## Number of Fisher Scoring iterations: 10
```

```r
test_model(firth_model, test)
```

```
##         MSPE     Error
## 1 0.07447477 0.1063333
```

As shown in the summary of the Firth model there are still estimates with large magnitudes however the standard error for these estimates have decreased which tells us that it just has a strong effect on our data. For example the p value of previous loan defaults was large because of it's standard error, however now it is near 0. However Firth Regression doesnt handle multi-collinearity like ridge regression thus we can still test for multi-collinearity which we will do utilizing VIF.

Multi-Collinearity (VIF)

The two variables with deviances greater than 10 were age and employment experience. After comparing the deviance of both, removing employment experience resulted in a lower deviance thus we chose to remove that variable. After that removal there were no more VIF > 10.

```r
vif(firth_model)
deviance(firth_model)
firth_without_age <- glm(
  formula = loan_status ~ . - person_age, data = train_no_outlier, family = "binomial",
  na.action = na.fail, method = "brglmFit"
)
firth_without_emp_exp <- glm(
  formula = loan_status ~ . - person_emp_exp, data = train_no_outlier, family = "binomial",
  na.action = na.fail, method = "brglmFit"
)
vif(firth_without_emp_exp)
# STOP no more VIF less than 10 or 5
```

```r
deviance(firth_without_age)
```

```
## [1] 12983.09
```

```r
deviance(firth_without_emp_exp)
```

```
## [1] 12977.64
```

```r
summary(firth_without_emp_exp)
```

```
##
## Call:
## glm(formula = loan_status ~ . - person_emp_exp, family = "binomial",
##     data = train_no_outlier, na.action = na.fail, method = "brglmFit")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25677  -0.12250  -0.00254  -0.00048   2.31803
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.295e+01  1.481e+00  -8.742  < 2e-16 ***
## person_age                       1.723e-02  7.104e-03   2.426   0.0153 *
## person_gendermale                4.756e-02  4.426e-02   1.075   0.2825
## person_education.L              -8.761e-02  1.218e-01  -0.720   0.4718
## person_education.Q              -7.896e-02  1.046e-01  -0.755   0.4503
## person_education.C              -1.097e-01  7.417e-02  -1.479   0.1391
## person_education^4              -7.227e-02  4.916e-02  -1.470   0.1415
## person_income                    3.575e-02  2.047e-02   1.747   0.0807 .
## person_home_ownershipOWN        -2.781e+00  1.251e-01 -22.229  < 2e-16 ***
## person_home_ownershipMORTGAGE   -9.202e-01  5.069e-02 -18.155  < 2e-16 ***
## person_home_ownershipOTHER      -6.936e-01  4.012e-01  -1.729   0.0839 .
## loan_amnt                       -8.563e-01  3.217e-02 -26.618  < 2e-16 ***
## loan_intentMEDICAL               6.787e-01  7.310e-02   9.284  < 2e-16 ***
## loan_intentEDUCATION            -1.529e-01  7.563e-02  -2.022   0.0432 *
## loan_intentVENTURE              -4.650e-01  8.131e-02  -5.719 1.07e-08 ***
## loan_intentHOMEIMPROVEMENT       1.103e+00  8.492e-02  12.994  < 2e-16 ***
## loan_intentDEBTCONSOLIDATION     1.081e+00  7.576e-02  14.263  < 2e-16 ***
## loan_int_rate                    4.564e-01  9.084e-03  50.240  < 2e-16 ***
## loan_percent_income              2.126e+01  4.257e-01  49.951  < 2e-16 ***
## cb_person_cred_hist_length      -2.645e-02  1.116e-02  -2.370   0.0178 *
## credit_score                    -1.185e-02  5.188e-04 -22.846  < 2e-16 ***
## previous_loan_defaults_on_fileNo  1.128e+01  1.439e+00   7.836 4.64e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36984  on 35495  degrees of freedom
## Residual deviance: 12978  on 35474  degrees of freedom
## AIC:  13022
##
## Type of estimator: AS_mixed (mixed bias-reducing adjusted score equations)
## Number of Fisher Scoring iterations: 11
```

```r
test_model(firth_without_emp_exp, test)
```

```
##        MSPE     Error
## 1 0.0745102 0.1066667
```

Stepwise AIC Variable Selection on Firth Logistic Regression

Forward Forward Stepwise AIC resulted in the original firth_model, no variabels were removed.

```
AIC.f <- stepAIC(firth_model, direction = "forward", k = 2, trace = FALSE)
test_model(AIC.f, test)
```

```
##         MSPE     Error
## 1 0.07447477 0.1063333
```

Backward Backward Stepwise AIC resulted in the removal of gender, education, and income variables. Gender, Education, Income

```
AIC.b <- stepAIC(firth_model, direction = "backward", k = 2, trace = FALSE)
test_model(AIC.b, test)
```

```
##         MSPE Error
## 1 0.07443141 0.106
```

Both Similar to Backward Stepwise AIC, Alternating Stepwise AIC resulted in the same removals as Backward Stepwise AIC.

```
AIC.both <- stepAIC(firth_model, direction = "both", k = 2, trace = FALSE)
test_model(AIC.both, test)
```

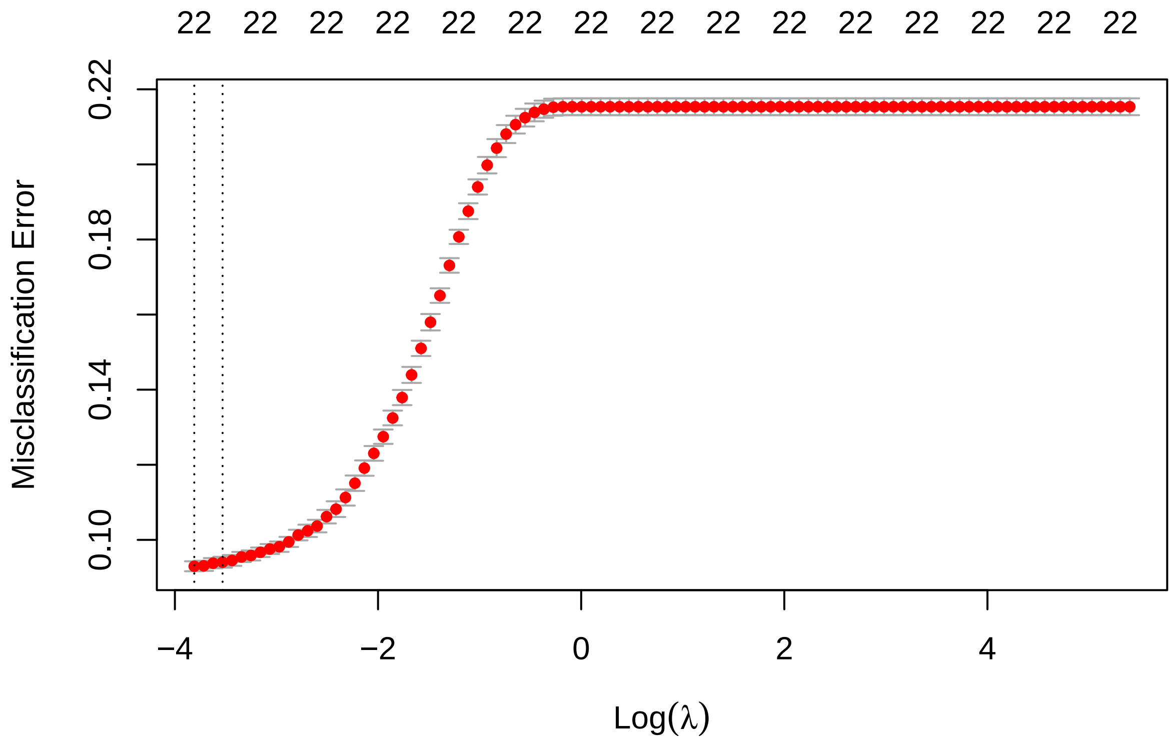
```
##         MSPE Error
## 1 0.07443141 0.106
```

Ridge Logistic Regression

The goal of ridge regression is to exchange increased bias for decreased variance. Ridge regression takes the ordinary least squares method and adds a penalty to it which minimizes the sum of squared residuals (RSS) plus the penalty, sum of coefficients (B_j)) squared times lambda, where lambda determines the severity of the penalty. This results in a smaller coefficients as we can in the below R code. In the case of Logistic Regression, Ridge Regression minimizes the sum of the likelihoods plus the sum of coefficients squared times lambda. But how do we choose a value of lambda that doesnt minimize and the coefficients too much but just enough? We use Cross-Validation, specifically 10-fold-cross validation to determine which lambda results in the lowest SSE. Another useful thing about Ridge Regression is that we do not need to do any variable selection as Ridge Regression already minimizes coefficients that would be removed from various variable selection methods.

```
x_train <- model.matrix(loan_status ~ ., data = train_no_outlier)[, -1]
y_train <- train_no_outlier$loan_status

cv_ridge <- cv.glmnet(
  x_train, y_train,
  family = "binomial",
  alpha = 0,
  nfolds = 10,
  type.measure = "class"
)
plot(cv_ridge)
```




```r
lambda_min <- cv_ridge$lambda.min # lambda that minimises CV error
lambda_1se <- cv_ridge$lambda.1se
lambda_min
```

```
## [1] 0.02216404
```

```r
ridge_model <- glmnet(
  x_train, y_train,
  family = "binomial",
  alpha = 0,
  lambda = lambda_min
)

coef(ridge_model)
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                                       s0
## (Intercept)                 -4.591604682
## person_age                   0.002185293
## person_gendermale            0.015019924
## person_education.L          -0.034690058
## person_education.Q           0.007919038
## person_education.C          -0.019048617
## person_education^4          -0.022281856
## person_income               -0.315897557
## person_emp_exp              -0.005405100
## person_home_ownershipOWN    -1.390395863
## person_home_ownershipMORTGAGE -0.750843115
## person_home_ownershipOTHER  -0.171525294
## loan_amnt                   -0.164841899
## loan_intentMEDICAL           0.336736920
```

```
## loan_intentEDUCATION            -0.223927016
## loan_intentVENTURE              -0.370796006
## loan_intentHOMEIMPROVEMENT       0.497792185
## loan_intentDEBTCONSOLIDATION     0.558104462
## loan_int_rate                    0.263097147
## loan_percent_income              9.640888494
## cb_person_cred_hist_length      -0.004235670
## credit_score                    -0.004942970
## previous_loan_defaults_on_fileNo  2.842972175
```

```r
x_train <- model.matrix(loan_status ~ ., data = train_no_outlier)[, -1]

x_test <- model.matrix(loan_status ~ ., data = test)[, -1]

x_test <- x_test[, colnames(x_train)]

p_hat <- predict(ridge_model,
  newx = x_test,
  s = "lambda.min",
  type = "response"
)[, 1]
pred_class <- ifelse(p_hat > 0.5, 1, 0)

data.frame(
  MSPE = mean((test$loan_status - p_hat)^2),
  Error = 1 - mean(pred_class == test$loan_status)
)
```

```
##        MSPE     Error
## 1 0.0767909 0.1051111
```

Why Not LASSO

We decided to use Ridge regression instead of LASSO regression soley because of the fact we wanted to keep all of our coefficients even if they aren't impactful as they still provide some sense of inference.

Analysis & Results

The different regression methods we used were Ordinary Logistic Regression, Firth Logistic Regression, and Ridge Regression. We applied different variable selection techniques such as Stepwise AIC and Mallow Cp.

Ordinary Logistic Regression

- The logistic full model (loan_status ~ .) with the original training data.
- The logistic null model (loan_status ~ 1) with the original training data and Drop-in Deviance test.
- The logistic model (loan_status ~ .) with the omitted outliers training data.

Firth's Logistic Regression

- The firth logistic model (loan_status ~ . - person_emp_exp) with the omitted outliers training data
- The firth logistic model (loan_status ~ . - person_emp_exp) with the omitted outliers training data and variable selection using VIF and StepwiseAIC

Ridge Logistic Regression

- The Ridge logistic model (loan_status ~ .) with the ommitted outliers and with 10-fold cross-validation.

Results

To evaluate predictive performance we applied each fitted model to the 20% test data and recorded the mean-squared prediction error (MSPE) and the classification error rate at a 0.5 cut-off.

Model (training setup)	**MSPE**	**Prediction Error Rate**
Ordinary Logistic Regression		
Full model, original data	0.07304491	0.1061111
Null model, Drop-in-Deviance test	0.1729012	0.2223333
Full model, outliers removed	0.07450024	0.1063333
Firth's Logistic Regression		
Firth, outliers removed	0.07447477	0.1063333
Firth (– person_emp_exp), outliers removed	0.0745102	0.1066667
Firth + VIF + Stepwise AIC	0.07443141	0.106
Ridge Logistic Regression		
Ridge (10-fold CV, λ_{min}), outliers removed	0.0767909	0.1051111

Conclusion

Best-performing model

Based off of prediction error rate, which is the most important to us for this analysis, ridge regression produced the lowest error rate and thus was our best performing model.

Handling separation

During the study we encountered both perfect and quasi-complete separation—most prominently for the "previous loan defaults" variable. Ordinary logistic regression produced huge, unstable coefficients and warnings. Firth's bias-reduced logistic regression resolved this by adding the a penalty, giving us more interpretable estimates.

Effect of variable selection

Stepwise AIC on the Firth model removed predictors that contributed little additional deviance, reducing model complexity without decreasing accuracy. Ridge regression, however, implicitly performs continuous shrinkage, retaining all variables and avoiding stepwise search.

Practical insight

Separation and multicollinearity require different remedies. Firth correction fixes the former, while ridge regression tackles the latter. In our loan-approval data, we can assume multicollinearity was the stronger driver of error, thus ridge emerged as the overall winner.

Sources

- https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-complete-or-quasi-complete-separation-in-logistic-regression-and-what-are-some-strategies-to-deal-with-the-issue/

- https://medium.datadriveninvestor.com/firths-logistic-regression-classification-with-datasets-that-are-small-imbalanced-or-separated-49d7782a13f1