Loan Applicant Analysis with Credit Data

Spencer Yee

Table of Contents

Abstract

Identifying a potential bad borrower depends on a loaner's ability to estimate the probability of having bad credit and plays a huge role when it comes to a loaner's decision to lend money to potential applicants. This study examines a dataset of 900 modern loan applicants. Logistic regression models were fit to the binary response for good or bad applicant. Modified variables for year of birth and applicant income are found to be important determinants of such. The methods used and final prediction function will be useful in making financial business decisions.

Introduction

The ability to take out a loan, no matter the purpose, is a very common and important aspect of everyday life. There are numerous factors that go towards analyzing loan applicants and their credit scoring data not limited to; number of dependents, income and outgoings on mortgage, other loans and credit cards. However these variables can potentially be used to predict whether a potential loan applicant will be a "good" or "bad" customer. This is extremely important as it can help to ensure that loans are not freely distributed to high-risk applicants who may be incapable of repayment. From a business standpoint, this is incredibly important as it is crucial that banks or other loaners can gauge the ability for a customer to repay a loan. In addition they can use the predicted results to determine whether they need to investigate a potential high-risk borrower further. To put it more simply, loaners are more likely to lend to a low-risk borrower with good credit than a high-risk borrower with low credit. This credit scoring data can help to distinguish between the two and make predictions of customer standing. This study examines a credit

scoring dataset of loan applicants born anytime within the 20th century. It contains the necessary features and content in order to conduct proper analysis and produce a predictive model. Outgoings on other moneylendings, income and dependents typically affect credit score and are likely to have an effect in determining a "good" vs "bad" customer. All other variables will also be investigated to determine if their effects are significant enough to be included in the model. The organization for the rest of the report is as follows. Section 2 will contain some characteristics of the data, section 3 will delve into model selection and analysis, and section 4 will summarize and conclude the report.

Data Characteristics & Exploration

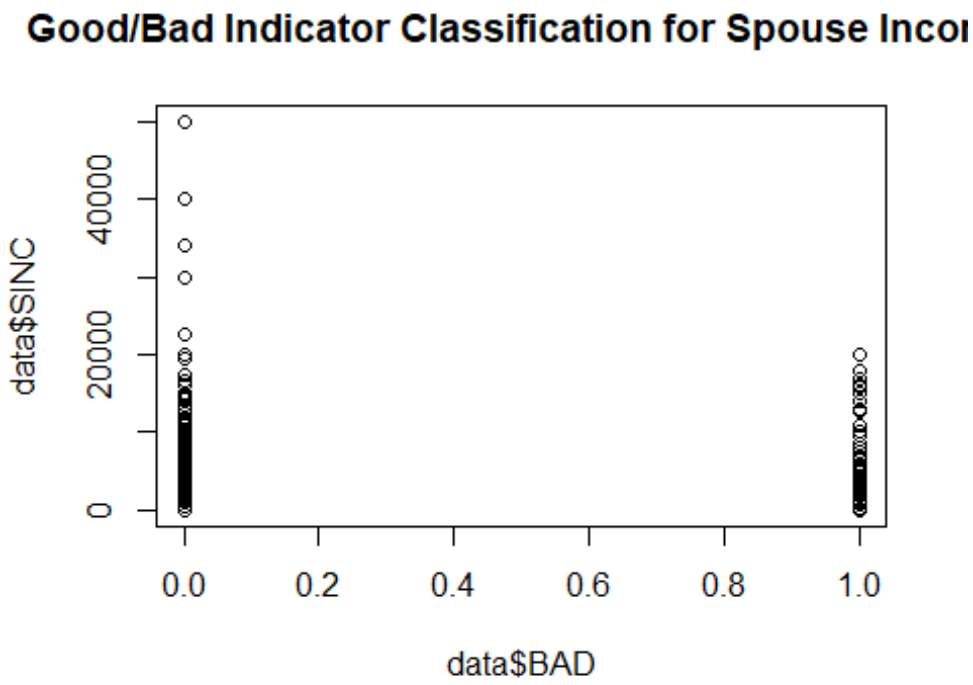
```
setwd("C:/Users/Spencer Yee/Desktop/MA380/AssignmentTWO")
data = read.table("credit-data-train.txt")
library(magrittr)
library(dplyr)
library(forcats)
library(ggplot2)
```

The data are cross-sectional and part of the book *Credit Scoring and Its Applications* by by Lyn C. Thomas, David B. Edelman, and Jonathan N. Crook. It is based on credit data taken on loan applicants born within the 20th century. The outcome of interest is the good/bad indicator which is a binary categorical variable where 1 = "bad" and 0 = "good". We have 14 other variables describing family, residential, and financial characteristics. There are 900 total observations. The following table shows the variables available and their definitions.

Item	Variable	Definition
1	DOB	Year of birth
2	NKID	Number of children
3	DEP	Number of other dependents
4	PHON	Is there a home phone (1 = yes, 0 = no)
5	SINC	Spouse's income
6	AES	Applicant's employment status (11 levels)
7	DAINC	Applicant's income
8	RES	Residential status (5 levels)
9	DHVAL	Value of home
10	DMORT	Mortgage balance outstanding
11	DOUTM	Outgoings on mortgage or rent
12	DOUTL	Outgoings on loans
13	DOUTHP	Outgoings on hire purchase
14	DOUTCC	Outgoings on credit cards
15	BAD	Good/bad indicator (1 = Bad, 0 = Good)

For year of birth (DOB) the year is abbreviated to omit the '19' (i.e 59 –> 1959). It should also be noted that if unknown, the year will read as 99. The data possesses 4 observations with such a value and as the next highest year is 1969, it is safe to assume all four are unknown. These variables may be removed if DOB is deemed significant. Spouse income (SINC) is skewed right due to the fact that many applicants appear to have spouse's with 0 income. This likely indicates that a majority of applicants have no spouse or do not work, which is expected. In the plot below we can see that there are some observations with values above 30,000 that may be considered outliers however it is apparent that all have repaid their loans. This is notable as it may indicate that spouse income may be a good predictor of a low vs. high-risk borrower. This characteristic of having outliers will thus be ignored.

```
plot(data$BAD, data$SINC, main = "Good/Bad Indicator Classification for
Spouse Income")
```



For employment status (AES) we have eleven employment categories labeled B, E, M, N, P, R, T, U, V, W, Z. These are nominal and are coded as [V = Government, W = housewife, M = military, P = private sector, B = public sector, R = retired, E = self employed, T = student, U = unemployed, N = others, Z = no response]. There are six observations which contain "Z" for no response. Applicant's income (DAINC) appears skewed right due to 160 applicants reporting 0 income. This is slightly strange, however disregarding this value presents relatively normally distributed values. Perhaps the reason that these applicants are applying for a loan is because they possess no income. For residential status (RES) we have five residential categories labeled F, N, O, P, U. These are also nominal and coded as [O = Owner, F = tenant furnished, U = Tenant Unfurnished, P = With parents, N = Other, Z = No response]. It should be noted that although 'Z' indicates no response, no observations

contain this value thus decreasing the levels to five. For home value (DHVAL) and mortgage balance (DMORT), they contain coding for unique responses and are coded as [0 = no response or not owner, 000001 = zero value, blank = no response]. Other values are randomly distributed integers. Neither of the variables contain blank values as evident by the output below which shows FALSE for the argument of each observation. There are also no observations with zero value (000001) for either variable.

```
idx1 = data$DHVAL == ""
table(idx1)

## idx1
## FALSE
##   900

idx2 = data$DMORT == ""
table(idx2)

## idx2
## FALSE
##   900
```

Home value (DHVAL) contains 476 observations with value 0 and mortgage balance (DMORT) contains 529. The minimum number of "no response" out of these observations can be estimated in table 1 of observations which contain "O" for residential status (RES) and 0 for home value (DHVAL). A true positive would indicate an applicant stated they were the owner of a home in RES but recorded 0 (no response or not owner) for home value. We will assume that an applicant did not enter false information for one of the variables and instead chose to not respond for home value. Thus the table below shows 36 true positives indicating that at least 36 applicants decided to not respond. The other 440 applicants are not owners.

```
idx1 = data$RES == "O"
idx2 = data$DHVAL == 0
table(idx1, idx2)

##        idx2
## idx1    FALSE TRUE
##   FALSE    10  440
##   TRUE    414   36
```

The same process for mortgage balance (DMORT) and outgoings on mortgage (DOUTM) can be used to estimate the minimum amount of no responses for mortgage balance. If outgoings (DOUTM) contains a value above 0, it would contradict a mortgage balance value of 0 indicating "not owner". There are 250 true positives indicating the minimum observations with no response and the remaining 279 as not owner.

```
idx1 = data$DMORT == 0
idx2 = data$DOUTM > 0
table(idx1, idx2)
```

```
##        idx2
## idx1    FALSE TRUE
##   FALSE   116  255
##   TRUE    279  250
```

The four variables pertaining to outgoings contain randomly distributed integers however all four variables possess a vast majority of observations with values of 0. It should be noted that the means of the numeric variables that which contain many values of 0 are dragged down as a result of the aforementioned.

The overall BAD credit rate (failing to pay loans) is calculated to be 0.267. As this variable is binary, it is easy to determine that the true values for applicants who both fail and succeed in paying back loans is 240 and 660 respectively. Using the bootstrap method, we can also use this mean to calculate a confidence interval for the mean BAD credit rate for all loan applicants. The output below indicates that we can be 95% confident that this rate will fall within 0.238 and 0.296. We should expect a potential logistic model's predictions to have a BAD credit rate that falls within this interval. We can use this when estimating the skill and fit of a model. We can also determine whether a variable is useful for a model predicting credit by seeing if BAD credit rate deviates from this interval when comparing to the response variable.

```
set.seed(12563)
N <- 10000
b.bad.rate <- numeric(N)
i <- 1
for(i in 1:N) {
  b.sample <- sample(data$BAD, nrow(data), replace = TRUE)
  b.bad.rate[i] <- mean(b.sample)
  i <- i + 1
}
m <- mean(b.bad.rate)
q <- quantile(b.bad.rate, probs = c(0.025, 0.975))
round(c(q[1], "Mean" = m, q[2]), 3)
```

```
##  2.5%  Mean 97.5%
## 0.238 0.267 0.296
```

Next, we can consider possible transformations of the given variables in order to potentially reduce the amount of predictors included in our model.

For year of birth (DOB), this variable can be converted into a new variable called age which will retain the same statistical meaning while providing a more comprehensible denotation. As this data was taken from a novel published in 2002, it can be assumed that this data was recorded in few years prior. We will assume the year 2000 for mathematical simplicity. Thus age can be defined as 2000 - (1900 + DOB). In addition, a new binary variable will be created to indicate if age is unknown by finding applicants with age = 1 (remember 99 indicates unknown year of birth). This will be used to remove unknown values during variable testing to prevent error.

```
data$age <- 2000 - (1900 + data$DOB)
ind.age <- which(data$age == 1)
data$age[ind.age] <- 0
data$age.unkn <- rep(0, length(data$age))
data$age.unkn[ind.age] <- 1
```

Looking at number of kids (NKID) and number of dependents (DEP) it is reasonable to assume that combining these two variables can make for one single variable for all dependents. We will then create another categorical variable for dependents that contains one level for 0 dependents and another level for more than one (1+) dependent. Likewise to dependents, the four variables pertaining to outgoings can be also be combined to create a single variable for total outgoings. An indicator variable will also be produced to show applicants with 0 total outgoings.

```
data$all.dep <- data$NKID + data$DEP
data$depend <- factor(ifelse(data$all.dep < 1, "0", "1+"), levels = c("0",
"1+"))
data$outgoings = data$DOUTM + data$DOUTL + data$DOUTHP + data$DOUTCC
ind.out <- which(data$outgoings == 0)
data$out.unkn <- rep(0, length(data$outgoings))
data$out.unkn[ind.out] <- 1
```

For the two income variables of applicant income (DAINC) and spouse income (SINC), we will again combine the two for a single total income variable. We will also assume that if total income = 0, income is unknown and another binary indicator variable will be created for income = 0 and income > 0. Like for age, this will be used to remove unknown values during testing. Interestingly, this can lead us to produce a new variable for disposable income which can be calculated by subtracting the annual outgoings from annual income. It would make sense that this variable could be a good predictor of credit as one would expect a portion of disposable income to be used to repay loans.

```
data$income <- data$DAINC + data$SINC
ind.inc <- which(data$income == 0)
data$inc.unkn <- rep(0, length(data$income))
data$inc.unkn[ind.inc] <- 1
data$disposable <- pmax(data$income - 12 * data$outgoings, 0)
```

Another considered transformation for both income variables was to perform “binning”, or to convert the two numerical variables into factor variables with multiple levels based on a range of reference values. The idea here is that it will be easier to recognize patterns in income within these smaller sub levels.

```
data <- data %>%
  mutate(ap.inc = case_when(
    income == 0 ~ "Zero",
    income <= 9300 ~ "(0,9300]",
    income <= 13700 ~ "(9300,13700]",
    income <= 17000 ~ "(13700,17000]",
    income <= 19500 ~ "(17000,19500]",
    income <= 22500 ~ "(19500,22500]",
```

```
    income <= 25500 ~ "(22500,25500]",
    income <= 30500 ~ "(25500,30500]",
    income <= 37000 ~ "(30500,37000]",
    income <= 45000 ~ "(37000,45000]",
    income <= Inf ~ "(45000,Inf]"))
data$ap.inc <- factor(data$ap.inc,
                      levels =
c("Zero","(0,9300]","(9300,13700]","(13700,17000]",

"(17000,19500]","(19500,22500]","(22500,25500]",

"(25500,30500]","(30500,37000]","(37000,45000]",
                                 "(45000,Inf]"))

data <- data %>%
  mutate(sp.inc = case_when(
    SINC == 0 ~ "Zero",
    SINC <=  1800 ~ "(0,1800]",
    SINC <=  2200 ~ "(1800,2200]",
    SINC <=  3000 ~ "(2200,3000]",
    SINC <=  5000 ~ "(3000,5000]",
    SINC <=  6000 ~ "(5000,6000]",
    SINC <=  7500 ~ "(6000,7500]",
    SINC <=  9500 ~ "(7500,9500]",
    SINC <= 11000 ~ "(9500,11000]",
    SINC <= 15000 ~ "(11000,15000]",
    SINC <= Inf ~ "(15000,Inf]"))
data$sp.inc <- factor(data$sp.inc,
                      levels = c("Zero","(0,1800]","(1800,2200]","(2200,3000]",
                                 "(3000,5000]","(5000,6000]","(6000,7500]",
                                 "(7500,9500]","(9500,11000]","(11000,15000]",
                                 "(15000,Inf]"))
```

It is clear that residential and employment status contain a rather large amount of levels so it was explored to see whether certain levels could be combined by viewing the data tables for each level. 23, 89, 14, 3, 394, 86, 85, 7, 164, 29, 6 This table shows us that some have an abnormally small number of observations, and it is understandable relative to their coding. As a result, we will combine B, M, N, U, W and Z and label this as “Other”. The other levels will remain the same, but a the new variable with the combined “other” level will be renamed. For residential status, there appears to be nothing wrong with observations per category so we will instead explore a logical reason to combine certain levels. We can see that there are three levels that can be classified as “Tenant” which are F, U and P. A new variable will be created to account for this.

```
data$emp.stat <- fct_collapse(data$AES, "Other" = c("B", "M", "N", "U", "W",
"Z"))
data$house.stat <- fct_collapse(data$RES, "Tenant" = c("F", "U", "P"))
```

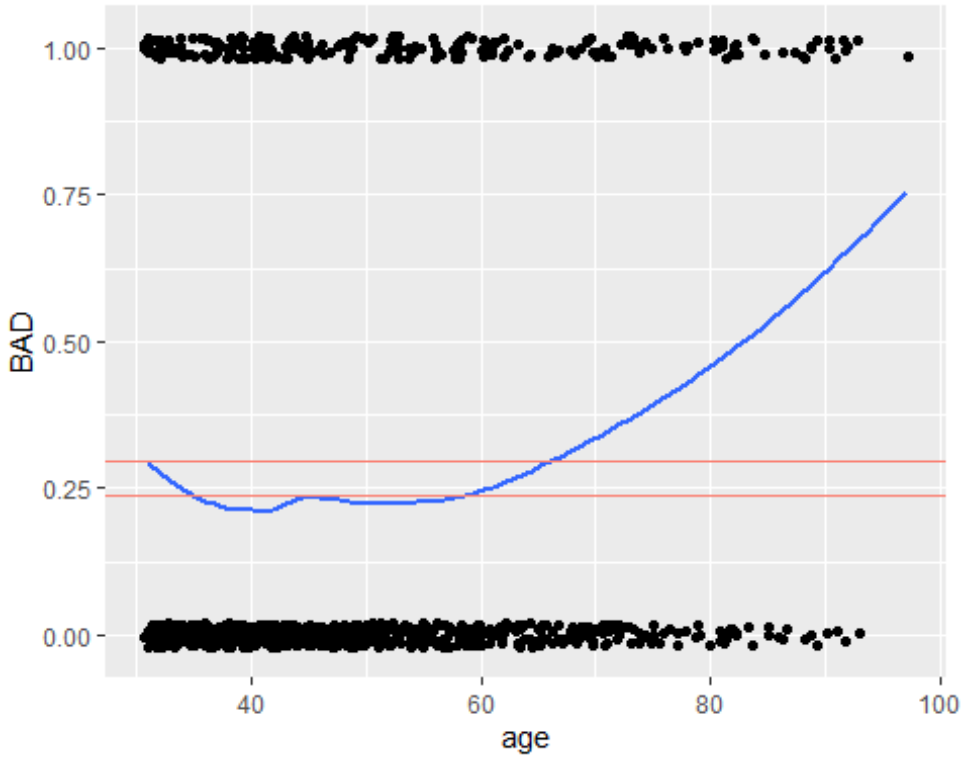
After producing these transformed variables, we can explore whether any of these variables show association to the response variable by producing either tables or ggplots against the BAD variable. It should be noted that the original variables should also be explored. An example table and ggplot for the new variables of dependents and age respectively.

```
tbl <- rbind(tapply(data$BAD, data$depend, sum),
             tapply(1-data$BAD, data$depend, sum),
             tapply(data$BAD, data$depend, mean))
dimnames(tbl)[[1]] <- c("BAD", "GOOD", "%BAD")
tbl
```

```
##                 0          1+
## BAD  161.0000000  79.0000000
## GOOD 439.0000000 221.0000000
## %BAD   0.2683333   0.2633333
```

```
ggplot(data[-ind.age,]) +
  aes(x = age, y = BAD) +
  geom_jitter(height = 0.02) +
  geom_smooth(se = FALSE) +
  geom_hline(yintercept = q, col = "salmon")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



We can see from the table that dependents shows no probability variation from the expected mean and thus is not a significant predictor. However, the ggplot for age shows an

increase in probability of having good credit as age also increases. There also appears to be slight quadratic curvature indicating we should also consider a polynomial variable of age.

```
data$agesq = (data$age)^2
```

These two methods of exploration were conducted for all variables and it was determined that age, age^2, applicant income (DAINC), spouse income (SINC), total income, home value cubed (DHVAL^3), total outgoings and disposable income could be useful predictors for good/bad credit. It was also found that possible interaction terms for total income and income indicator, and total outgoings and outgoings indicator.

The next section will discuss the building of potential models using these variables and ultimately a single model selection.

Model Selection & Interpretation

Section 2 determined that there is relevant association between credit rating and the loan applicant characteristics variables. It also described how some of these variables could be transformed to provide better predictors for a logistic regression model. This model is intended to be included in a function that can be useful in predicting whether we should consider a loan applicant "good' or"bad". This section summarizes the methods used in selecting the best model and will interpret the final prediction function and its results in context. Finally, this section will touch base on the utilization and benefits of this function from an analytic business standpoint.

Preliminarily, there were fourteen potential models that were considered, each consisting of different potentially significant components, but each fitted with maximum likelihood. However, this was narrowed down to five by using the Hosmer-Lemeshow Test of Goodness Fit. The composition for these models (labeled m4, m5, m6, m7, m9) are as follows:

```
m4 <- glm(BAD ~ agesq + income + out.unkn,
          data = data,
          family = binomial(link = "logit"))

m5 <- glm(BAD ~ agesq + ap.inc,
          data = data,
          family = binomial(link = "logit"))

m6 <- glm(BAD ~ agesq + ap.inc + outgoings,
          data = data,
          family = binomial(link = "logit"))

m7 <- glm(BAD ~ agesq + ap.inc + out.unkn,
          data = data,
          family = binomial(link = "logit"))

m9 <- glm(BAD ~ age + agesq + ap.inc,
```

```
          data = data,
          family = binomial(link = "logit"))
```

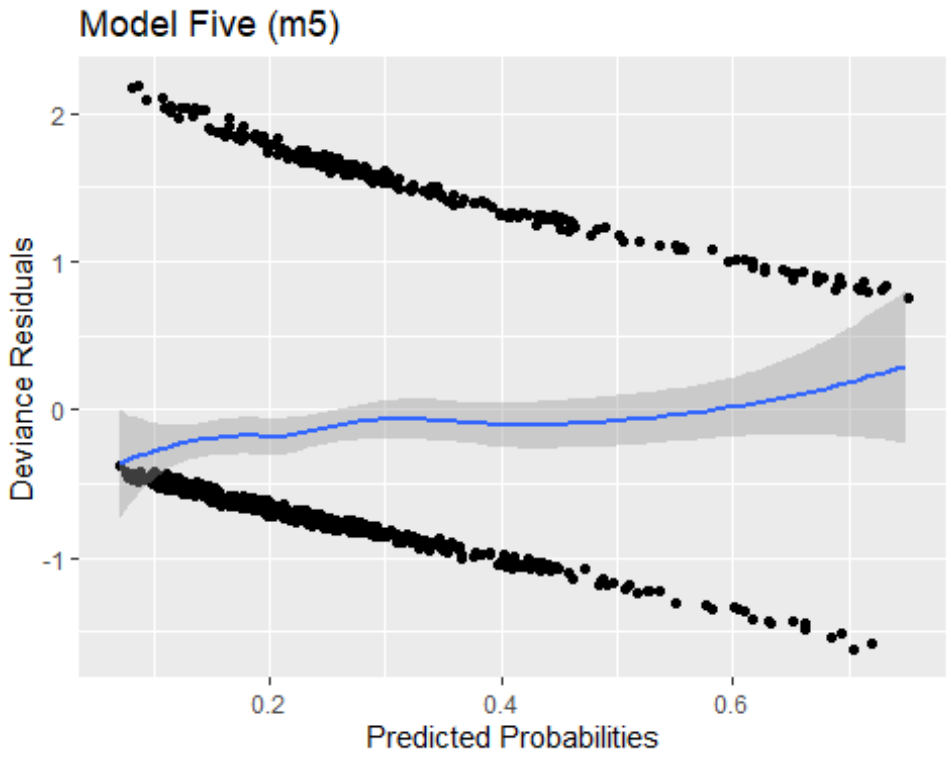
Next, a table of classification metrics for the five models was produced using a preliminary threshold of 0.45 to seek high performance values. As we are specifically attempting to pinpoint the bad applicants, we must pay more attention to sensitivity. The table for both HL test and metrics can be found below:

Model	HL Stat.	P-Value	Precision	Accuracy	Sensitivity	Specificity
m4	2.312264	0.9699114	55.00	74.22	18.33	94.55
m5	2.531471	0.9602472	55.56	74.44	20.83	93.94
m6	2.628238	0.9554826	54.55	74.22	20.00	93.94
m7	2.659839	0.9538604	54.05	74.33	25.00	92.27
m9	2.984993	0.9352958	55.17	74.33	20.00	94.09

From this table it is hard to distinguish a single best model. While the HL test suggests that model four (m4) would be the best, the model also produces the lowest sensitivity. We now have to ask ourselves questions such as if a 0.01 decrease in p-value in model five (m5) is worth an increase of 2.50 for sensitivity. We can attempt to visualize these differences by fitting a plot for each model's predicted probabilities against its deviance residuals. Again, as this is a binary response variable, we are seeking to find which model's plot displays the flattest (horizontal) smoothing line (blue). We will also add a little noise to the plot to see results more clearly. The described plot for models five and six can be seen here:

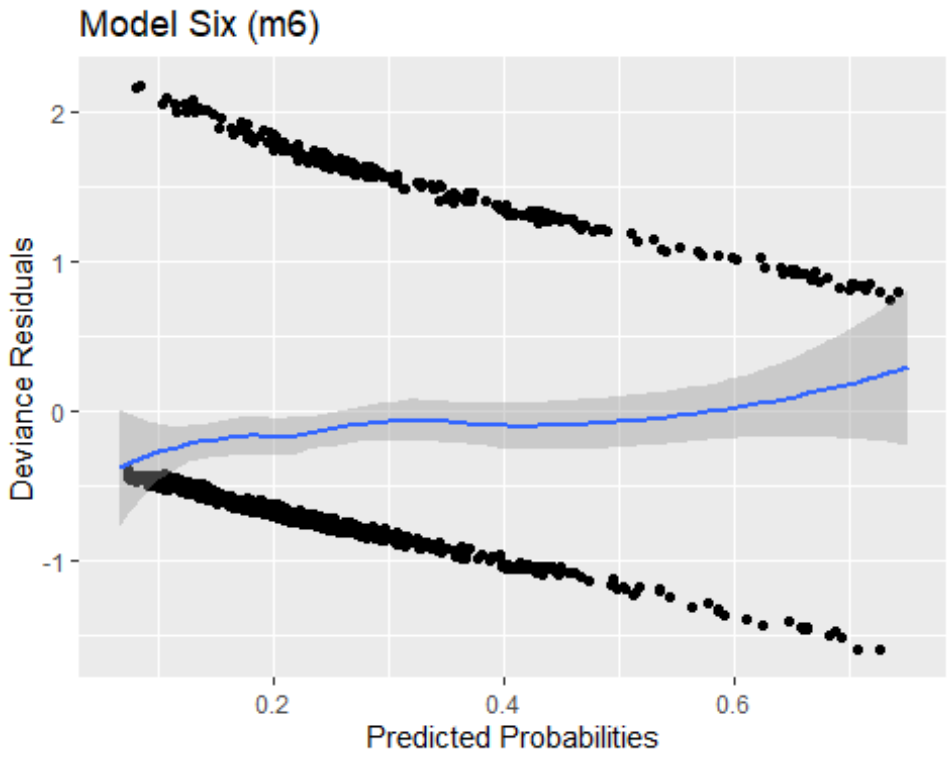
```
p <- predict(m5, type = "response")
r <- resid(m5, type = "deviance")
ggplot(data = data.frame(x = p, y = r)) +
  aes(x = x, y = y) +
  geom_jitter(height = 0.02, width = 0.02) +
  geom_smooth() +
  labs(title = "Model Five (m5)", x = "Predicted Probabilities", y =
"Deviance Residuals")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
p <- predict(m6, type = "response")
r <- resid(m6, type = "deviance")
ggplot(data = data.frame(x = p, y = r)) +
  aes(x = x, y = y) +
  geom_jitter(height = 0.02, width = 0.02) +
  geom_smooth() +
  labs(title = "Model Six (m6)", x = "Predicted Probabilities", y = "Deviance
Residuals")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



As visualized above, there is unrecognizable difference between the performance of the two models to distinguish a superior, yet I will note that they noticeably outperformed models four, seven and nine. I will subsequently rule those out and shift focus towards five and six. After re-evaluating the aforementioned table, I will select model five, as it produces a better test statistic and classification metrics compared to model six. To reiterate, the systematic components include the transformed age-squared and binned applicant income variables. The model's coefficients are displayed below:

```
coef(m5)
```

```
##         (Intercept)                agesq      ap.inc(0,9300] 
##        -0.504238705         0.000170173          -0.726955583 
##  ap.inc(9300,13700] ap.inc(13700,17000] ap.inc(17000,19500] 
##        -0.951648020         -0.672938068          -1.726268706 
## ap.inc(19500,22500] ap.inc(22500,25500] ap.inc(25500,30500] 
##        -1.604890017         -0.875763896          -1.307003500 
## ap.inc(30500,37000] ap.inc(37000,45000]    ap.inc(45000,Inf] 
##        -0.945984043         -1.312812997          -2.078267597
```

To briefly describe the functionality of this model, the respective coefficients for age-squared and each income level represent the effect of those variables on the odds ratio for our dependent variable. Thus, our positive coefficient for age-squared indicates that a one unit change in this variable will increase the probability of an applicant having bad credit. As for the categorical income levels, these negative coefficients will indicate the opposite. The finalized odds ratio produced by this model ultimately determines the probability of an applicant being a "bad" customer. However as we are more interested in this model

generating a definitive output, we can set a probability threshold which will essentially define applicants with odds ratio above the threshold as bad and those below it as good. I will choose a threshold of 0.45 (45%).

The K-fold cross validation method will be utilized to further test this model's performance against random training samples. This will allow us to determine if we have overfit our data. I will select k = 10 to test the model against 10 training sets and then compare the average of each folds metrics to the whole dataset. The results can be viewed below:

```
cm.metrics <- function(cm) {
  acc <- sum(diag(cm))/sum(cm)
  pre <- cm[1,1]/sum(cm[1,])
  sen <- cm[1,1]/sum(cm[,1])
  spe <- cm[2,2]/sum(cm[,2])

  ans <- c("Accuracy" = acc,
           "Precision" = pre,
           "Sensitivity" = sen,
           "Specificity" = spe)
  return(ans)
}

M <- matrix(NA, nrow = 5, ncol = 4)
l <- list(m4, m5, m6, m7, m9)
i <- 1
for(f in l){
  p <- predict(f, type = "response")
  PC <- ifelse(p > 0.45, 1, 0)
  TC <- data$BAD
  cm <- table(factor(PC, levels = 1:0),
              factor(TC, levels = 1:0))
  M[i,] <- cm.metrics(cm)
  i <- i + 1
}

set.seed(324632)
data$fold <- sample(c(rep(0, 90), rep(1, 90), rep(2, 90),
                      rep(3, 90), rep(4, 90), rep(5, 90),
                      rep(6, 90), rep(7, 90), rep(8, 90), rep(9, 90)),
                    nrow(data),
                    replace = FALSE)

set.seed(398490)
F <- matrix(NA, nrow = 10, ncol = 5) # for storing our metrics for each fold
dimnames(F)[[2]] <- c("fold", "accuracy", "precision", "sensitivity",
"specificity")
i <- 1
for(fld in 0:9){
  fit <- glm(BAD ~ agesq + ap.inc,
```

```
              data = data,
              subset = fold != fld, # do not use one fold
              family = binomial(link = "logit"))

  p <- predict(fit,
               newdata = subset(data, subset = fold == fld), # on the fold
not used
               type = "response")
  PC <- ifelse(p > 0.45, 1, 0)
  TC <- data$BAD[data$fold == fld] # True condition on the fold predicted
  cm <- table(factor(PC, levels = 1:0),
              factor(TC, levels = 1:0))
  F[i,] <- c(fld, cm.metrics(cm))
  i <- i + 1
}
(fld.means <- apply(F, 2, mean)[2:5]) # calculate means for each column

##    accuracy   precision sensitivity specificity
##   0.7400000   0.5445887   0.1924348   0.9358238

whole.sample <- M[2,]
tbl <- rbind("Whole Sample" = whole.sample,
             "Cross-Validated" = fld.means,
             "Difference" = whole.sample - fld.means)
dimnames(tbl)[[2]] <- c("Accuracy", "Precision", "Sensitivity",
"Specificity")
round(tbl,4)

##                 Accuracy Precision Sensitivity Specificity
## Whole Sample      0.7478    0.5823      0.1917      0.9500
## Cross-Validated   0.7400    0.5446      0.1924      0.9358
## Difference        0.0078    0.0377     -0.0008      0.0142
```

From these results we can see that the model performs very well and we should expect no concerns continuing with this model. I will now create a predictive function called “score” which will use this model to take a dataset and predict whether a loan applicant would potentially make for a bad customer.

```
score <- function(newdata){
  data <- newdata

  data$age <- 2000 - (1900 + data$DOB)
  ind.age <- which(data$age == 1)
  data$age[ind.age] <- 0
  data$age.unkn <- rep(0, length(data$age))
  data$age.unkn[ind.age] <- 1
  data$agesq <- (data$age)^2

  data <- data %>%
    mutate(ap.inc = case_when(
```

```
      income == 0 ~ "Zero",
      income <= 9300 ~ "(0,9300]",
      income <= 13700 ~ "(9300,13700]",
      income <= 17000 ~ "(13700,17000]",
      income <= 19500 ~ "(17000,19500]",
      income <= 22500 ~ "(19500,22500]",
      income <= 25500 ~ "(22500,25500]",
      income <= 30500 ~ "(25500,30500]",
      income <= 37000 ~ "(30500,37000]",
      income <= 45000 ~ "(37000,45000]",
      income <= Inf ~ "(45000,Inf]"))
  data$ap.inc <- factor(data$ap.inc,
                        levels =
c("Zero","(0,9300]","(9300,13700]","(13700,17000]",

"(17000,19500]","(19500,22500]","(22500,25500]",

"(25500,30500]","(30500,37000]","(37000,45000]",
                                  "(45000,Inf]"))

  data <- data %>%
    mutate(sp.inc = case_when(
      SINC == 0 ~ "Zero",
      SINC <=  1800 ~ "(0,1800]",
      SINC <=  2200 ~ "(1800,2200]",
      SINC <=  3000 ~ "(2200,3000]",
      SINC <=  5000 ~ "(3000,5000]",
      SINC <=  6000 ~ "(5000,6000]",
      SINC <=  7500 ~ "(6000,7500]",
      SINC <=  9500 ~ "(7500,9500]",
      SINC <= 11000 ~ "(9500,11000]",
      SINC <= 15000 ~ "(11000,15000]",
      SINC <= Inf ~ "(15000,Inf]"))
  data$sp.inc <- factor(data$sp.inc,
                        levels =
c("Zero","(0,1800]","(1800,2200]","(2200,3000]",
                                  "(3000,5000]","(5000,6000]","(6000,7500]",

"(7500,9500]","(9500,11000]","(11000,15000]",
                                  "(15000,Inf]"))



  p <- predict(m5, newdata = data, type = "response")
  ans <- ifelse(p > 0.45, 1, 0)
  return(ans)
}
```

The ability to predict bad credit is highly beneficial for loaners, yet there is not substantial collateral for a borrower to fail to repay a loan. In addition, although loaner profitability is certainly a significant factor when issuing loans, there are ways to compensate for the risks of lending to bad customers such as higher interest rates. This will lead to more revenue. However this is only the case if a customer is actually capable of repaying the entirety of given loan. This concept is why a variation of income was included in the model and directly found to be significant. Our probability threshold of 0.45 for determining bad credit is most reasonable because it maximizes accuracy and precision.

Summary and Concluding Remarks

It is obvious that all loan applicants and their credit scores are unique. However it is possible to establish significant determinants of credit score. The best-fitted logistic model concluded that bad credit score can be predicted by the twice transformed year of birth variable, and also applicant income after being split into specific ranges of values. This was motivated by evidence that these transformed variables better fit the variation in response variable data and that fewer explanatory variables would provide more accurate predictions.

This study was based on 900 loan applicants born within the 20th century and contained sufficient variables relating to applicant characteristics that may prove to influence credit score. This allowed us to build and test numerous models in order to determine the best for predicting bad credit.

The prediction function created in this report should be directly applicable to a majority of loan applicants. Although the model's ability to correctly predict true bad credit score is low, it is at the very least incredibly useful at correctly predicting applicants with good credit score. This in turn will vastly lessen the pool of applicants that which require further manual examination for loan consideration.

References

Crook, Jonathan N., David B. Edelman and Lyn C. Thomas, **Credit Scoring and Its Applications,** 2002, SIAM.

Appendix