Imperial College London

COURSEWORK 1

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

Consumer Credit Risk Modelling

Author: 01844579

Date: December 4, 2022

1 Data Preparation for Model A

In order to build our first logistic regression model we will analyse each variable and will process each accordingly. First, we check if there are any *NA* values and find that *emp_length_p* has 7974 *NA* values, which we will assume as being unemployed so we will set the *NA* values to 0. *loan_amnt* is skewed with value 0.6977016 and so in the interest of a robust model we would use the square root which results in a skewness of 0.1147523. However, taking the logarithm gives a skewness of -0.6913015 but presents a lower p-value and so we will choose this transformation, as discussed in lectures. Next, we have the *grade* variable which is discrete and so we will need to process it using weight of evidence. Next we have *term* which we will assume only takes values 36 and 60 and therefore convert to levels using as.factor(). *addr_state* is a categorical variable with 50 values and therefore we will use weight of evidence to process it.

2 Summary of Model A

Below shows the summary of Model described in Question 4 built using the training set.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.639299   0.152968  17.254  < 2e-16 ***
loan_amnt   -0.090619   0.016641  -5.446 5.16e-08 ***
grade       -1.047762   0.019434 -53.913  < 2e-16 ***
emp_length_p 0.026083   0.002559  10.194  < 2e-16 ***
term60       0.176759   0.024797   7.128 1.02e-12 ***
addr_state  -0.965668   0.078857 -12.246  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 76772  on 104722  degrees of freedom
Residual deviance: 72836  on 104717  degrees of freedom
AIC: 72848

Number of Fisher Scoring iterations: 5
```

Looking at the summary from Question 4 we see that all of our predictor variables are stastically significant at a level of 0.01. First, *loan_amnt* has a negative association with non-defaulting. However, we see a positive association with non-defaulting for *emp_length_p*. Additionally, we see a positive association of having a *term* of 60 relative to a *term* of 36. Since *grade* and *addr_state* have been transformed using

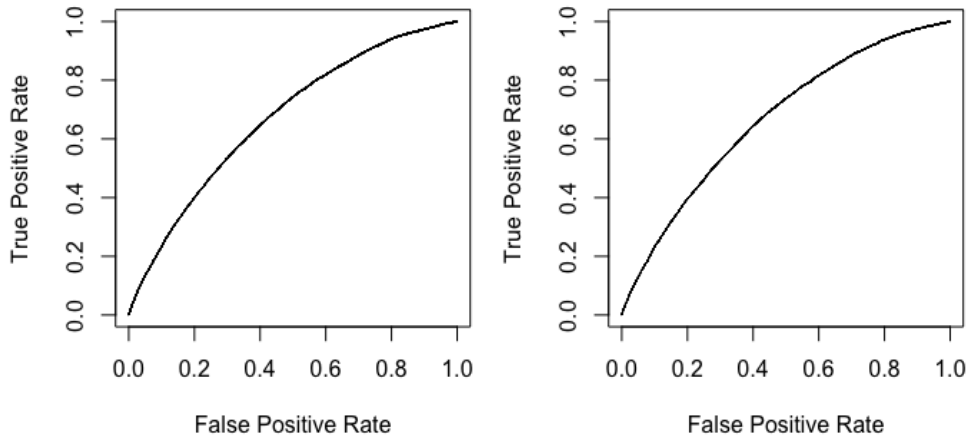
weight of evidence and so we can not conclude the affect on non-defaulting without the weight of evidence value for each grade and state.

3 Performance of Model A

Using code from the notes we obtain our ROC curves, demonstrated in Figure 1 and received an AUC score for the training set of 0.6700031 and an AUC score for the testing set of 0.6666141. The AUC score for the training set suggests an under-fitted model and since we obtain a similar score for the test set it suggests our model does not over-fit the data.

4 Data Processing for Final Model

We will inherit the processed variables from the previous model. First, we will transform *annual_inc* since it has a skewness of 26.92386 and using a logarithmic transform the skewness becomes 0.2059106. Next we will use weight of evidence on *purpose_p* since creating 10 indicator variables wouldn't promote a parsimonious model and would create inefficient coefficient estimates. *revol_bal* has a skewness of 18.34403 and so in the interest of a robust model we will add 1 to the variable and take the logarithm, resulting in a skewness of -2.130593. For *avg_cur_bal* there are 5 *NA* values which we will assume as not having any accounts and therefore set equal to 0. Lastly, we will use weight of evidence on *issue_d* as, once again, having 12 indicator variables is not feasible. The rest of the variables we will not process since the *glm* package will automatically treat the variable accordingly and skewness was



(a) ROC of Training Data (b) ROC of Test Data

Figure 1: ROC Curves

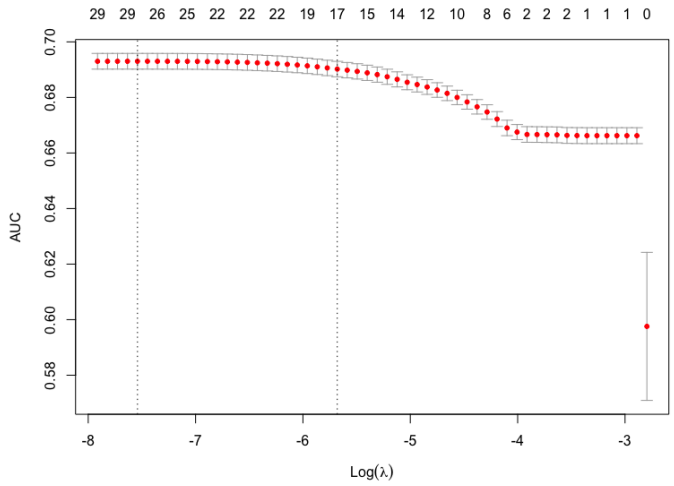


Figure 2: Cross-validation Error of Lambda

used as a processing diagnostic in the interest of robustness since logistic regression is able to handle skewed data appropriately.

5 Lasso Penalty for Variable Selection

We will perform LASSO penalty for variable selection instead of stepwise selection due to the large data size. Motivated by a parsimonious model we will use LASSO regression over ridge regression because we want to shrink coefficients to 0. To visualise the optimal lambda that minimises the cross-validation error of lambda we will plot it, shown in Figure 2.

```
Measure: AUC

      Lambda Index Measure       SE Nonzero
min 0.000531    52  0.6930 0.002833      28
1se 0.003412    32  0.6902 0.002776      16
```

After analysing the performance we will choose lambda.1se which is the largest λ such that the error is within one standard error of the cross-validated errors for the minimum λ instead of lambda.min since the AUC is only lower by 0.0028 and only uses 17 variables as oppose to 28 variables. Using the selected variables from our LASSO penalty we can now build a logistic regression with these variables.

6 Building Model B

From the 34 available variables LASSO penalty has returned: *loan_amnt, int_rate, grade, emp_length_p, annual_inc, revol_bal, avg_curl_bal, dti, inq_last_6mths,*

mo_sin_old_rev_tl_op, mo_sin_rcnt_rev_tl_op, mo_sin_rcnt_tl, mort_acc, num_actv_rev_tl, addr_state, issue_d. So we will build a logistic regression with these variables. However after building the model,

```
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            2.153e+00  3.034e-01   7.096 1.28e-12 ***
loan_amnt             -2.157e-01  1.946e-02 -11.087  < 2e-16 ***
int_rate              -5.701e-02  7.347e-03  -7.759 8.56e-15 ***
grade                 -4.501e-01  5.522e-02  -8.151 3.60e-16 ***
emp_length_p           1.415e-02  2.876e-03   4.919 8.70e-07 ***
annual_inc             2.276e-01  2.856e-02   7.971 1.57e-15 ***
avg_cur_bal            5.487e-06  9.990e-07   5.492 3.97e-08 ***
dti                   -1.339e-02  1.387e-03  -9.651  < 2e-16 ***
inq_last_6mths        -6.864e-02  9.551e-03  -7.186 6.67e-13 ***
mo_sin_old_rev_tl_op   1.209e-03  1.321e-04   9.156  < 2e-16 ***
mo_sin_rcnt_tl         1.331e-02  1.878e-03   7.086 1.38e-12 ***
mort_acc               3.948e-02  6.117e-03   6.454 1.09e-10 ***
num_actv_rev_tl       -8.116e-03  3.642e-03  -2.229   0.0258 *
addr_state            -8.698e-01  8.182e-02 -10.631  < 2e-16 ***
issue_d               -1.019e+00  1.673e-01  -6.090 1.13e-09 ***
revol_bal              1.673e-06  7.729e-07   2.164   0.0304 *
mo_sin_rcnt_rev_tl_op  1.975e-03  9.484e-04   2.083   0.0373 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 71907  on 99403  degrees of freedom
Residual deviance: 67098  on 99387  degrees of freedom
AIC: 67132

Number of Fisher Scoring iterations: 5
```

we see that at the 1% significance level *num_actv_rev_tl, revol_bal* and *mo_sin_rcnt_rev_tl_op* are statistically insignificant and therefore we will remove them from our model. So our new model summary reads as:

```
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            2.027e+00  2.846e-01   7.122 1.06e-12 ***
loan_amnt             -2.126e-01  1.922e-02 -11.059  < 2e-16 ***
int_rate              -5.731e-02  7.344e-03  -7.803 6.04e-15 ***
grade                 -4.517e-01  5.519e-02  -8.183 2.76e-16 ***
emp_length_p           1.360e-02  2.865e-03   4.746 2.07e-06 ***
annual_inc             2.337e-01  2.750e-02   8.499  < 2e-16 ***
```

```
avg_cur_bal           6.605e-06  9.495e-07   6.956 3.50e-12 ***
dti                  -1.311e-02  1.315e-03  -9.966  < 2e-16 ***
inq_last_6mths       -7.051e-02  9.527e-03  -7.401 1.35e-13 ***
mo_sin_old_rev_tl_op  1.229e-03  1.301e-04   9.441  < 2e-16 ***
mo_sin_rcnt_tl        1.609e-02  1.546e-03  10.410  < 2e-16 ***
mort_acc              3.836e-02  6.103e-03   6.285 3.27e-10 ***
addr_state           -8.707e-01  8.179e-02 -10.644  < 2e-16 ***
issue_d              -1.011e+00  1.672e-01  -6.048 1.47e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 71907  on 99403  degrees of freedom
Residual deviance: 67112  on 99390  degrees of freedom
AIC: 67140

Number of Fisher Scoring iterations: 5
```

and all of our variables are statistically significant at the 1% level. In fact with this model we receive an AUC score of 0.6915052 for the training data and an AUC score of 0.6872728 for the test data an improvement from Model A.

7 Interaction Terms

The next step in improving Model B would be using interaction terms or a segmented model; in this report we will only be exploring interaction terms due to time constraints. There are two methods of interaction term selection, manual inspection and automated search. In an automated search we would be using a stepwise procedure and measuring the performance using AIC, however, due to the number of variables in Model B there are too many combinations even at order 2. So we will be selecting interaction terms by manual inspection, comparing the model with the interaction term with the model without interaction terms using a likelihood ratio test at a 1% significance level. When adding the interaction term *annual_inc * loan_amnt* we receive a p-value of 0.0005052116 and so the addition of the interaction term gives a better fit. However, if we also add the interaction term *grade * int_rate* we receive a p-value of 0 when compared to the model with just the one interaction term. When implementing the new model with the two interaction terms we received an AUC score of 0.6924734 for the training set and an AUC score of 0.6878926 for the test set. So after using the likelihood ratio test and comparing the AUC scores our final model will inlucde the interaction terms *annual_inc * loan_amnt* and *grade * int_rate*.

8 Summary

In building Model A we simply analysed, processed and transformed the data appropriately and fitted a logistic regression with these new processed variables. In building Model B we processed the rest of the variables and used LASSO penalty to reduce the number of variables used. After de-selecting some of the variables we checked the p-values once again to removed any statistically insignificant variables at the 1% level. To further improve the model after using LASSO penalty we tried adding interaction terms using manual inspection and comparing the models with the likelihood ratio test. By comparing AUC scores for both the training and testing set we see that Model B performs better, additionally the margin between scores for both training and testing is very small suggesting a model that is not over-fitted to the data. However the relatively low score as a whole suggests that Model B still rather discriminates non-defaulters poorly. In conclusion we were able to improve Model A by adding some more variables and interaction terms, however the increase in performance can be argued as marginal.

A Code

```
# import packages
library(moments)
library(glmnet)

# create functions
woe.tab <- function(x,y) {
  n1 <- sum(y)
  n0 <- sum(1-y)
  nx0n1 <- tapply(1-y,x,sum)*n1
  nx1n0 <- tapply(y,x,sum) *n0
  nx0n1[which(nx0n1==0)]<-n1
  nx1n0[which(nx1n0==0)]<-n0
  return(log(nx0n1)-log(nx1n0))
}

woe.assign <- function(wtab, x) {
  w<-rep(0,length(x))
  ni<-names(wtab)
  for (i in 1:length(ni)) {
    w[which(x==ni[i])]<-wtab[i]
  }
  return(w)
}

roc <- function(y, s)
{
```

```
  yav <- rep(tapply(y, s, mean), table(s))
  rocx <- cumsum(yav)
  rocy <- cumsum(1 - yav)
  area <- sum(yav * (rocy - 0.5 * (1 - yav)))
  x1 <- c(0, rocx)/sum(y)
  y1 <- c(0, rocy)/sum(1 - y)
  auc <- area/(sum(y) * sum(1 - y))
  print(auc)
  plot(x1,y1,"l", xlab ='False Positive Rate', ylab= 'True Positive
   ↪  Rate')
}


# create outcome
names(D1)[1] <- 'non_def'
D1$non_def <- as.numeric(D1$non_def == 'FALSE')

# check for NA values
#sapply(D1, function(x) sum(is.na(x)))

# data processing/transformation for both Model A and B
D1$grade <- woe.assign(woe.tab(D1$grade, D1$non_def), D1$grade)
D1$term <- as.factor(D1$term)
D1$purpose_p <- woe.assign(woe.tab(D1$purpose_p, D1$non_def),
 ↪  D1$purpose_p)
D1$addr_state <- woe.assign(woe.tab(D1$addr_state, D1$non_def),
 ↪  D1$addr_state)
D1$issue_d <- woe.assign(woe.tab(D1$issue_d, D1$non_def), D1$issue_d)
D1$emp_length_p[is.na(D1$emp_length_p)] <- 0
D1$loan_amnt <- log(D1$loan_amnt)
D1$annual_inc <- log(D1$annual_inc)
D1$avg_cur_bal[is.na(D1$avg_cur_bal)] <- 0
D1$revol_bal <- log(D1$revol_bal + 1)


# create train and test data
ix <- sample(nrow(D1), 2/3 * nrow(D1), replace = FALSE)
D1train <- D1[ix,]
D1test <- D1[-ix,]

# build logistic regression for Model A
glm.4 <- glm(non_def ~ loan_amnt + grade + emp_length_p + term +
 ↪  addr_state, data = D1train, family=binomial("logit"))
summary(glm.4)
```

```
# predictions, ROC and AUC Model A
y1 <- predict(glm.4, D1train, type="link")
y2 <- predict(glm.4, D1test, type="link")
roc(D1train$non_def, y1)
roc(D1test$non_def, y2)

# LASSO penalty for variable selection
X <- data.matrix(D1[,-1])
Y <- data.matrix(D1[1])
Xtrain <- X[ix,]
Xtest <- X[-ix,]
Ytrain <- Y[ix,]
Ytest <- Y[-ix]
cv.lasso <- cv.glmnet(Xtrain, Ytrain, alpha = 1, family = "binomial",
 ↪  type.measure = "auc")
plot(cv.lasso)
model.min <- glmnet(Xtrain, Ytrain, alpha = 1, family = "binomial",
 ↪  lambda = cv.lasso$lambda.min)
model.1se <- glmnet(Xtrain, Ytrain, alpha = 1, family = "binomial",
 ↪  lambda = cv.lasso$lambda.1se)
assess.glmnet(model.min, newx = Xtest, newy = Ytest)$auc
assess.glmnet(model.1se, newx = Xtest, newy = Ytest)$auc
coef(cv.lasso, cv.lasso$lambda.min)
coef(cv.lasso, cv.lasso$lambda.1se)

# building model from LASSO penalty and after removing insignificant
 ↪  variables
model.1 <- glm(non_def ~ loan_amnt + int_rate + grade + emp_length_p +
 ↪  annual_inc
              + avg_cur_bal + dti + inq_last_6mths
              + mo_sin_old_rev_tl_op + mo_sin_rcnt_tl + mort_acc
               ↪  +addr_state +  issue_d,
              data = D1train,
              family = binomial("logit"))

# adding interaction terms
model.2 <- glm(non_def ~ annual_inc * loan_amnt + grade * int_rate +
 ↪  loan_amnt + int_rate + grade + emp_length_p + annual_inc
               + avg_cur_bal + dti + inq_last_6mths
               + mo_sin_old_rev_tl_op + mo_sin_rcnt_tl + mort_acc
                ↪  +addr_state +  issue_d,
               data = D1train,
               family = binomial("logit"))

# likelihood ratio test
```

```
1-pchisq(2*((-model.2$deviance) - (-model.1$deviance)),2)

# predictions, ROC and AUC for Model B
y1 <- predict(model.2, D1train, type="link")
y2 <- predict(model.2, D1test, type="link")
roc(D1train$non_def, y1)
roc(D1test$non_def, y2)
```