Code

Classification

Logistic Regression

Response in the logistic regression formula is the log odds of a binary outcome of 1. We only observe the binary outcome, not the log odds, so special statistical methods are needed to fit the equation. Logistic regression is a special instance of generalized linear model (GLM) developed to extend linear regression to other settings.

In R, to fit a logistic regression, glm function is used with family set to “binomial”. The following code fits a logistic regression to the personalloan data.

Hide

```
loan_data = read.csv("~/Dropbox/Priya-PhD- Documents/Courses/Data Analysis and Visualization-Spr
ing 2019/Datasets/loan_data.csv", header=TRUE)
head(loan_data)
```

```
  X      status loan_amnt      term annual_inc   dti payment_inc_ratio
1 1 Charged Off      2500 60 months      30000  1.00           2.39320
2 2 Charged Off      5600 60 months      40000  5.55           4.57170
3 3 Charged Off      5375 60 months      15000 18.08           9.71600
4 4 Charged Off      9000 36 months      30000 10.08          12.21520
5 5 Charged Off     10000 36 months     100000  7.06           3.90888
6 6 Charged Off     21000 36 months     105000 13.22           8.01977
  revol_bal revol_util            purpose home_ownership
1      1687        9.4                car           RENT
2      5210       32.6     small_business            OWN
3      9279       36.5              other           RENT
4     10452       91.7 debt_consolidation           RENT
5     11997       55.5              other           RENT
6     32135       90.3 debt_consolidation           RENT
  delinq_2yrs_zero pub_rec_zero open_acc grade outcome emp_length
1                1            1        3   4.8 default          1
2                1            1       11   1.4 default          5
3                1            1        2   6.0 default          1
4                1            1        4   4.2 default          1
5                1            1       14   5.4 default          4
6                1            1        7   5.8 default         11
            purpose_ home_  emp_len_ borrower_score
1     major_purchase  RENT  > 1 Year           0.65
2     small_business   OWN  > 1 Year           0.80
3              other  RENT  > 1 Year           0.60
4 debt_consolidation  RENT  > 1 Year           0.50
5              other  RENT  > 1 Year           0.55
6 debt_consolidation  RENT  > 1 Year           0.40
```

Hide

```
logistic_model = glm(outcome ~ payment_inc_ratio + purpose_ + home_ + emp_len_ + borrower_score,
data = loan_data, family = 'binomial' )
logistic_model
```

```

Call:  glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Coefficients:
               (Intercept)           payment_inc_ratio
                  -1.63809                    -0.07974
purpose_debt_consolidation    purpose_home_improvement
                  -0.24937                    -0.40774
    purpose_major_purchase             purpose_medical
                  -0.22963                    -0.51048
             purpose_other      purpose_small_business
                  -0.62066                    -1.21526
                  home_OWN                   home_RENT
                  -0.04833                    -0.15732
         emp_len_ > 1 Year              borrower_score
                   0.35673                     4.61264

Degrees of Freedom: 45341 Total (i.e. Null);  45330 Residual
Null Deviance:      62860
Residual Deviance: 57510    AIC: 57540
```

The response is outcome, which takes a 0 if the loan is paid off and 1 if the loan defaults. purpose_ and home_ are factor variables representing the purpose and the home ownership status. As in regression, a factor variable with P levels is represented using P-1 columns. By default, reference coding is used and the levels are all compared to the refernce level. The reference level for these factors are credit_card and MORTGAGE respectively. The variable borrower_score is a score from 0 to 1 (poor to excellent) representing the creditworthiness of the borrower.

Generalized Linear Models

GLMs are characterized by two main components: - A probability distribution or family (binomial in case of logistic regression) - A link function mapping the response to the predictors (logit in case of logistic regression)

Logistic regression is the most common form og GLM. Sometimes log link function is used instead of logit.Poisson distribution is used to model count data (number of times user visit a web pge in certain amount of time). Other families include negative binomial and gamma (often used to model elapsed time).

Predicted Values from Logistic Regression

The predicted value from logistic regression is in terms of log odds. $\hat{Y} = log(Odds(Y = 1))$.

Hide

```
pred = predict(logistic_model)
summary(pred)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
-3.509606 -0.505061  0.008539 -0.002564  0.518825  2.704774 
```

Converting these values to probabilities

Hide

```
prob = 1/(1 + exp(-pred))
summary(prob)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.02904 0.37635 0.50213 0.50000 0.62687 0.93731 
```

These are on a scale from 0 to 1 and don't indicate whether the predicted value is default or paid off. We could declare any value greater than 0.5 as default. A lower cutoff is often appropriate if the goal is to identify members of a rare class.

Interpreting the Coefficients and Odds Ratios

Odds ratio is guven by – $odds ratio =

This is interpreted as the odds that Y=1 when X=1 versus the odds that Y=1 when X=0. If the odds ratio is 2, then the odds that Y=1 are two times higher when X=1 versus X=0.

We work wit odds because the coeffcient β_j in the logistic regression is the log of the odds ratio for X_j.

Assessing the Model

Logistic regression is assessed by how accurately the model classifies new data.

Hide

```
summary(logistic_model)
```

```
Call:
glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.15528  -1.07421   0.05853   1.06908   2.51951

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -1.638092   0.073708 -22.224  < 2e-16 ***
payment_inc_ratio          -0.079737   0.002487 -32.058  < 2e-16 ***
purpose_debt_consolidation -0.249373   0.027615  -9.030  < 2e-16 ***
purpose_home_improvement   -0.407743   0.046615  -8.747  < 2e-16 ***
purpose_major_purchase     -0.229628   0.053683  -4.277 1.89e-05 ***
purpose_medical            -0.510479   0.086780  -5.882 4.04e-09 ***
purpose_other              -0.620663   0.039436 -15.738  < 2e-16 ***
purpose_small_business     -1.215261   0.063320 -19.192  < 2e-16 ***
home_OWN                   -0.048330   0.038036  -1.271    0.204
home_RENT                  -0.157320   0.021203  -7.420 1.17e-13 ***
emp_len_ > 1 Year           0.356731   0.052622   6.779 1.21e-11 ***
borrower_score              4.612638   0.083558  55.203  < 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: 62857  on 45341  degrees of freedom
Residual deviance: 57515  on 45330  degrees of freedom
AIC: 57539

Number of Fisher Scoring iterations: 4
```

Interpretation of p-value comes with the same caveat as in regression, and should be viewed more as a relative indicator of variable importance than a formal measure of statistical significance. Logistic regression model, which has a binary rfesponse, does not have an associated RMSE or R-squared. Logistic regression model is typically evaluated using more general metrics for classification.

Fit generalized additive model using “mgcv” package.

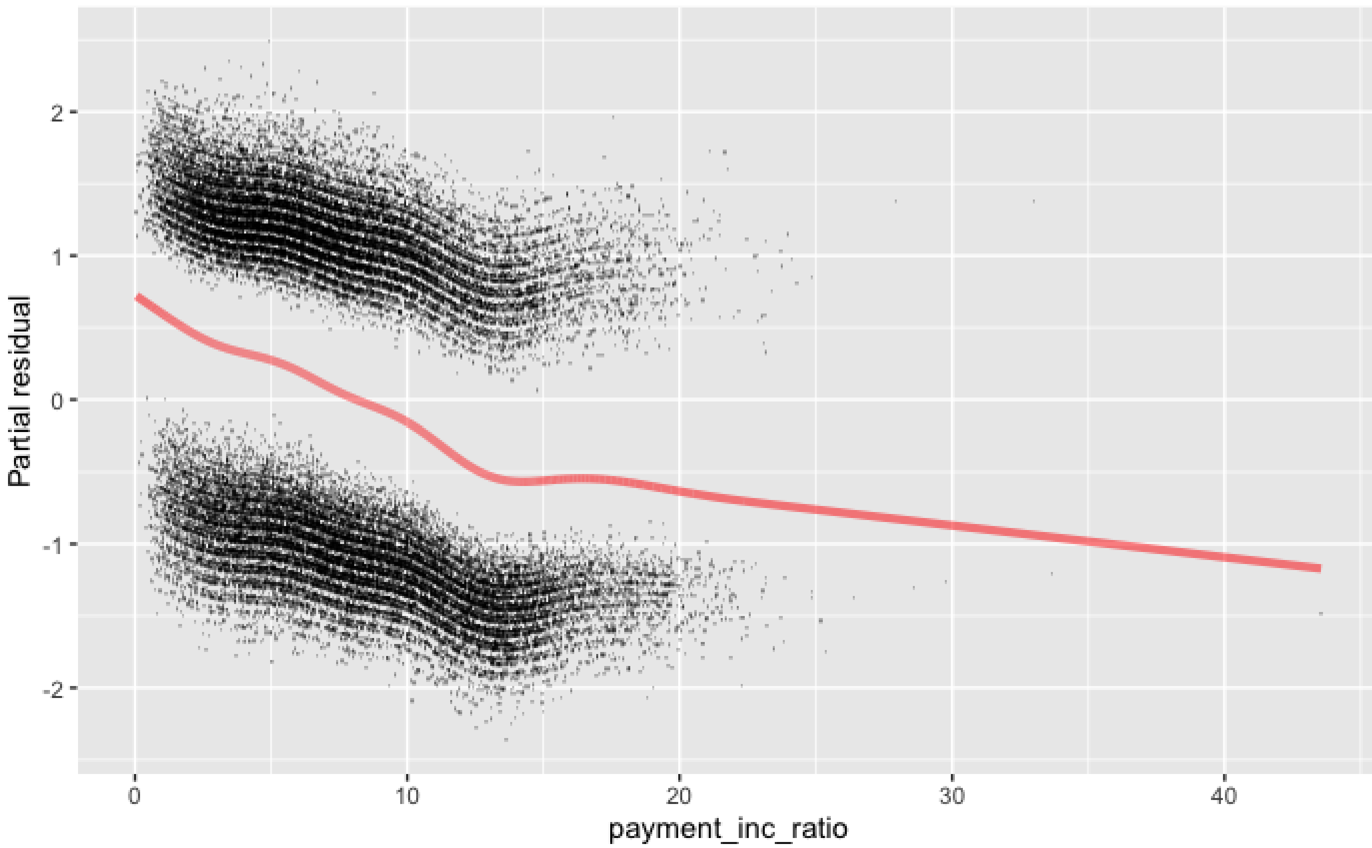
Hide

```
library(mgcv)
logistic_gam = gam(outcome~ s(payment_inc_ratio)+ purpose_ + home_ + emp_len_ + s(borrower_scor
e), data = loan_data, family = "binomial")
```

One area where logistic regression differs from linear regression is in the analysis of residuals.

Hide

```
terms = predict(logistic_gam, type = "terms")
partial_resid = resid(logistic_model) + terms
df = data.frame(payment_inc_ratio = loan_data[,'payment_inc_ratio'],
                terms = terms[,'s(payment_inc_ratio)'],
                partial_resid = partial_resid[,'s(payment_inc_ratio)'])
ggplot(df, aes(x = payment_inc_ratio, y = partial_resid, solid = FALSE)) +
  geom_point(shape = 46, alpha = .4)+
  geom_line(aes(x = payment_inc_ratio, y = terms), color ='red' , alpha = 0.5, size = 1.5) + lab
s(y = "Partial residual")
```



The estimated fit, shown by the line goes between two sets of point clouds. The top cloud correponds to a response of 1 (defaulted loans), and bottom cloud corresponds to a response of 0 (loans paid off). This is typical of residuals from a logistic regression since the output is binary. Partial residuals in logistic regression are useful to confirm non linear behavior and identify highly influential records.

Evaluating Classification Models

Applying holdout set approach

Hide

```
# Random sample indexes
train_index = sample(1:nrow(loan_data), 0.75 * nrow(loan_data))
test_index = setdiff(1:nrow(loan_data), train_index)
# Build train and test sets
train_set = loan_data[train_index, ]
test_set = loan_data[test_index, ]
```

Confusion Matrix

Hide

```
pred = predict(logistic_gam, newdata = train_set)
pred_y = as.numeric(pred > 0)
true_y =  as.numeric(train_set$outcome == "default")
true_pos = (true_y == 1) & (pred_y == 1)
true_neg = (true_y == 0) & (pred_y == 0)
false_pos = (true_y == 0) & (pred_y == 1)
false_neg = (true_y == 1) & (pred_y == 0)
conf_mat = matrix(c(sum(true_pos),sum(false_pos),
                    sum(false_neg),sum(true_neg)),2,2)
colnames(conf_mat) = c('Yhat = 1' ,'yhat = 0')
rownames(conf_mat) = c('Y = 1', 'Y = 0')
conf_mat
```

```
      Yhat = 1 yhat = 0
Y = 1     6300    10722
Y = 0    10927     6057
```

Hide

```
#precision
precision = conf_mat[1,1]/sum(conf_mat[,1])
precision
```

```
[1] 0.365705
```

Hide

```
#recall
recall = conf_mat[1,1]/sum(conf_mat[1,])
recall
```

```
[1] 0.3701093
```

Hide

```
#specificity
specificity = conf_mat[2,2]/sum(conf_mat[2,])
specificity
```

```
[1] 0.3566298
```

Hide

```
#FPR = 1 - specificity
```