Topic-7-MY OWN code Logistic Regression

Dawn Cheung

2024-04-29

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
churn = read.csv("~/Github/DSA1101 Slayers/datasets/churn.csv")

churn$Churned = as.factor(churn$Churned)
churn$Married = as.factor(churn$Married)
churn = churn[,-1]
attach(churn)

table(Churned)
```

```
## Churned
##    0    1
## 6257 1743
```

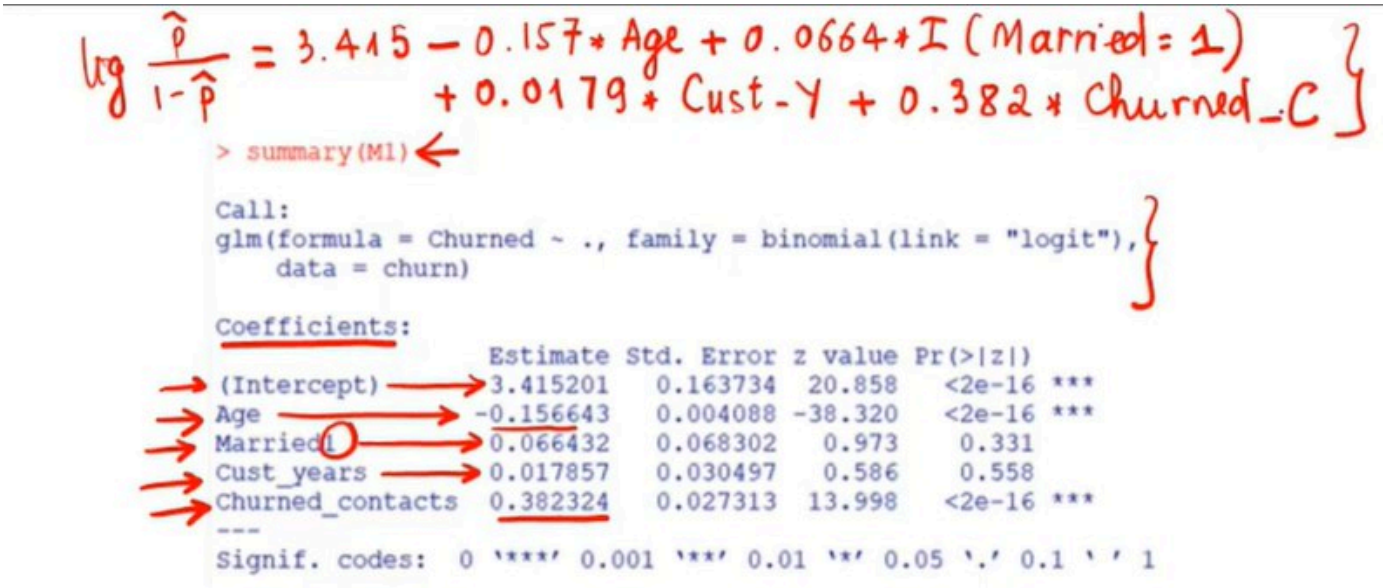
```
prop.table(table(Churned))
```

```
## Churned
##        0        1
## 0.782125 0.217875
```

Generalised Linear Model: glm()

```
M1 <- glm(Churned ~ . , # . is all columns except the response col
          data = churn,
          family = binomial(link = "logit")) #logit is default tho
#note that Churned must be equal to 0 and 1, it can be as.factor(), but the category names MU
ST be 0 or 1. Ironically doesnt matter if you converted to factor, for glm.

#MUST specify family = binomial, bc its a 1 or 0 catagory.
#otherwise, linear model will be formed
summary(M1)
```

```
##
## Call:
## glm(formula = Churned ~ ., family = binomial(link = "logit"),
##     data = churn)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.415201   0.163734  20.858   <2e-16 ***
## Age              -0.156643   0.004088 -38.320   <2e-16 ***
## Married1          0.066432   0.068302   0.973    0.331
## Cust_years        0.017857   0.030497   0.586    0.558
## Churned_contacts  0.382324   0.027313  13.998   <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: 8387.3  on 7999  degrees of freedom
## Residual deviance: 5357.9  on 7995  degrees of freedom
## AIC: 5367.9
##
## Number of Fisher Scoring iterations: 6
```



How to interpret coefficient of contact:

When the number of contacts increases by 1 unit, then the *log odds of churning* increases by [coefficient of contacts], keeping all other variables the same.

=> When the number of contacts increases by 1 unit, then the *odds of success changes by e^0.38 times* (or e^(coefficient of contacts) times), keeping all other variables the same

How to interpret married:

Comparing a married person (Married = 1) vs a non-married person, when other variables are the same, then the log odds of churning will be larger by 0.07.

=> Comparing a married person (Married = 1) vs a non-married person, then th odds of churning will change by e^0.07 times.

Remember that we want p-value to be as low as possible.

We see from the summary stats that the Pr(>|z|) value (ie the p value) for Cust_years is the highest, so lets drop it, and call the new model M2

```
M2 <- glm(Churned ~ Age + Married + Churned_contacts,
          data = churn,
          family = binomial) #logit is default

summary(M2)
```

```
##
## Call:
## glm(formula = Churned ~ Age + Married + Churned_contacts, family = binomial,
##     data = churn)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.472062   0.132107  26.282   <2e-16 ***
## Age              -0.156635   0.004088 -38.318   <2e-16 ***
## Married1          0.066430   0.068299   0.973    0.331
## Churned_contacts  0.381909   0.027302  13.988   <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: 8387.3  on 7999  degrees of freedom
## Residual deviance: 5358.3  on 7996  degrees of freedom
## AIC: 5366.3
##
## Number of Fisher Scoring iterations: 6
```

We see that the p value for married is still quiet large (0.331), so it is not that significant. Let's do another one lol

```
M3 <- glm(Churned ~ Age + Churned_contacts,
          data = churn,
          family = binomial) #logit is default

summary(M3)
```

```
##
## Call:
## glm(formula = Churned ~ Age + Churned_contacts, family = binomial,
##     data = churn)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.502716   0.128430   27.27   <2e-16 ***
## Age              -0.156551   0.004085  -38.32   <2e-16 ***
## Churned_contacts  0.381857   0.027297   13.99   <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: 8387.3  on 7999  degrees of freedom
## Residual deviance: 5359.2  on 7997  degrees of freedom
## AIC: 5365.2
##
## Number of Fisher Scoring iterations: 6
```

Note that M3 is built using train = full data set => the goodness of the model should also use the full dataset

Note that the intercept might be not be significant, but usually keep the intercept in the model, so dont remove it unless removing it makes sense in the data context.

ROC and AUC value for log model

log model calculates the probability of Y = 1 => more of a classifer Remember that AUC value is area under the ROC curve

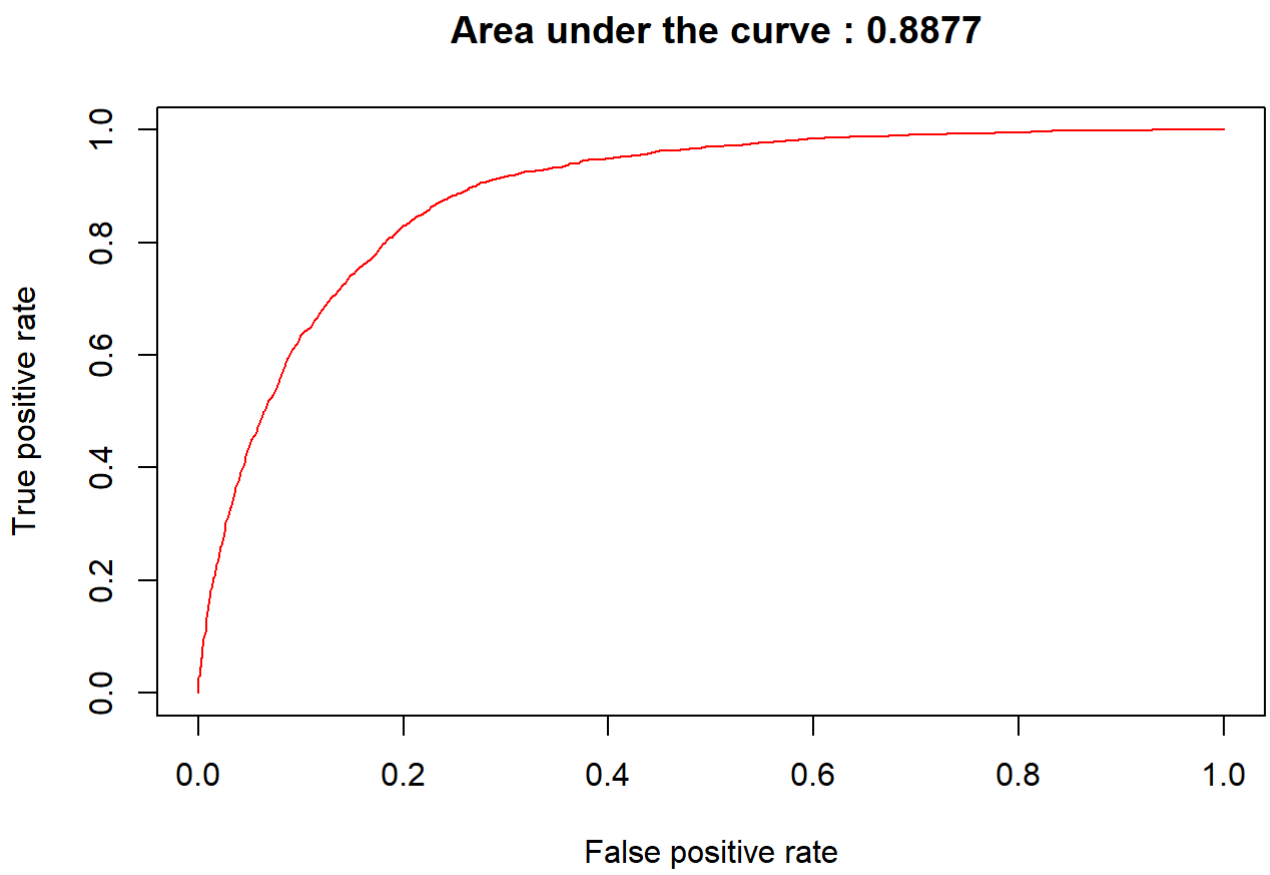
```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.3.3
```

```
#note model M3 is built using train = full dataset (row wise)
#so test the goodness of fit of the model by considering test = full dataset
prob = predict(M3, newdata = churn[,2:5], type = "response") #so newdata is just Age, Marrie
d, Churned. But honestly predict will just ignore Married cus M3 doesnt use Married. so no ne
ed to include it, its up to you.
#returns a vector of 8000 values

pred = prediction(prob, Churned)

roc = performance(pred, "tpr", "fpr")
auc = performance(pred, measure = "auc")

auc@y.values[[1]]
```

```
## [1] 0.8876509
```

```
plot(roc , col = "red", main = paste(" Area under the curve :", round(auc@y.values[[1]] ,4)))
```



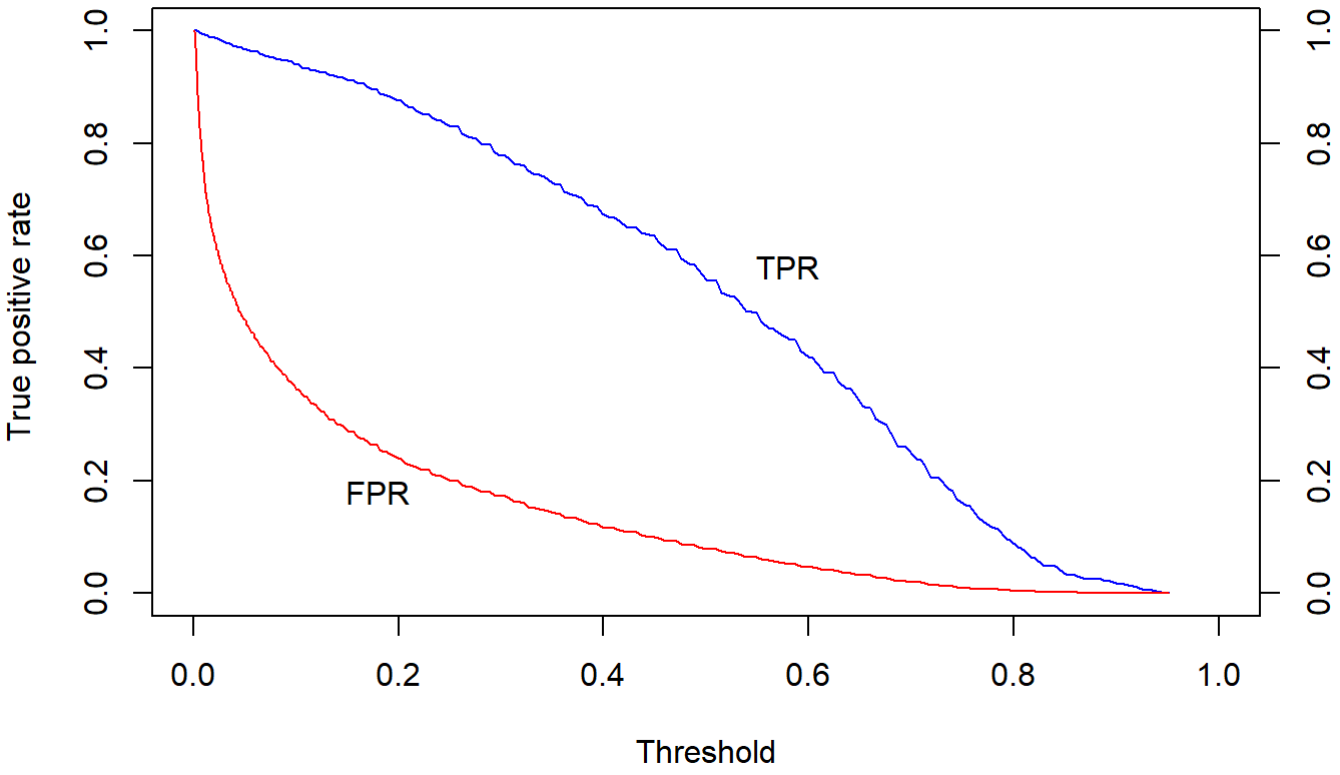
We should choose a lower threshold (than 0.5) when the success criteria is very rare ie a very low percentage of the population will be positive

how tpr and fpr changes when threshold changes

```
#step 1: extract the alpha(threshold), FPR and TPR values from roc (ie the performance(pred, "tpr", "fpr"))

alpha <- round(as.numeric(unlist(roc@alpha.values)) ,4)
length(alpha)
```

```
## [1] 328
```

```
fpr <- round(as.numeric(unlist(roc@x.values)) ,4)
tpr <- round(as.numeric(unlist(roc@y.values)) ,4)

plot(alpha, tpr ,xlab ="Threshold", xlim =c(0 ,1) ,
     ylab = "True positive rate ", type ="l", col = "blue")
par(new ="True")
plot(alpha, fpr, xlab ="", ylab ="", axes=F, xlim =c(0, 1) , type ="l", col = "red" )
axis(side =4) # to create an axis at the 4th side
mtext(side =4, line =3, "False positive rate")
text(0.18 ,0.18 , "FPR")
text(0.58 ,0.58 , "TPR")
```




```
# how tpr and fpr changes when threshold changes
# getting the precise best alpha value
cbind(alpha, tpr, fpr)[70:150,] #making sure the document isnt too long lol
```

```
##        alpha    tpr    fpr
##  [1,] 0.5826 0.4509 0.0523
##  [2,] 0.5705 0.4630 0.0563
##  [3,] 0.5658 0.4710 0.0577
##  [4,] 0.5611 0.4716 0.0582
##  [5,] 0.5536 0.4836 0.0603
##  [6,] 0.5489 0.4991 0.0639
##  [7,] 0.5441 0.4997 0.0639
##  [8,] 0.5394 0.5009 0.0642
##  [9,] 0.5318 0.5209 0.0689
## [10,] 0.5270 0.5267 0.0721
## [11,] 0.5223 0.5290 0.0724
## [12,] 0.5146 0.5347 0.0746
## [13,] 0.5099 0.5554 0.0793
## [14,] 0.5004 0.5571 0.0793
## [15,] 0.4927 0.5754 0.0831
## [16,] 0.4879 0.5841 0.0852
## [17,] 0.4832 0.5858 0.0853
## [18,] 0.4755 0.5944 0.0871
## [19,] 0.4708 0.6116 0.0921
## [20,] 0.4660 0.6116 0.0922
## [21,] 0.4613 0.6122 0.0925
## [22,] 0.4537 0.6254 0.0975
## [23,] 0.4490 0.6351 0.0997
## [24,] 0.4443 0.6368 0.1004
## [25,] 0.4367 0.6403 0.1028
## [26,] 0.4320 0.6495 0.1095
## [27,] 0.4274 0.6500 0.1096
## [28,] 0.4227 0.6500 0.1098
## [29,] 0.4152 0.6627 0.1130
## [30,] 0.4106 0.6684 0.1159
## [31,] 0.4060 0.6684 0.1165
## [32,] 0.3987 0.6741 0.1186
## [33,] 0.3941 0.6867 0.1235
## [34,] 0.3896 0.6885 0.1237
## [35,] 0.3850 0.6890 0.1243
## [36,] 0.3778 0.7034 0.1301
## [37,] 0.3733 0.7074 0.1331
## [38,] 0.3689 0.7080 0.1338
## [39,] 0.3618 0.7143 0.1355
## [40,] 0.3574 0.7263 0.1416
## [41,] 0.3530 0.7263 0.1424
## [42,] 0.3418 0.7407 0.1477
## [43,] 0.3375 0.7441 0.1506
## [44,] 0.3333 0.7447 0.1512
## [45,] 0.3265 0.7516 0.1541
## [46,] 0.3223 0.7613 0.1614
## [47,] 0.3182 0.7619 0.1616
## [48,] 0.3140 0.7625 0.1625
## [49,] 0.3075 0.7728 0.1691
## [50,] 0.3034 0.7774 0.1724
## [51,] 0.2994 0.7780 0.1726
## [52,] 0.2930 0.7854 0.1747
## [53,] 0.2891 0.7980 0.1798
## [54,] 0.2852 0.7986 0.1800
```

```
## [55,] 0.2813 0.7986 0.1803
## [56,] 0.2752 0.8090 0.1860
## [57,] 0.2714 0.8107 0.1895
## [58,] 0.2676 0.8118 0.1895
## [59,] 0.2617 0.8170 0.1926
## [60,] 0.2580 0.8290 0.1996
## [61,] 0.2544 0.8290 0.1998
## [62,] 0.2508 0.8290 0.2003
## [63,] 0.2451 0.8353 0.2054
## [64,] 0.2416 0.8405 0.2084
## [65,] 0.2381 0.8411 0.2090
## [66,] 0.2326 0.8451 0.2116
## [67,] 0.2292 0.8508 0.2191
## [68,] 0.2258 0.8520 0.2194
## [69,] 0.2225 0.8526 0.2196
## [70,] 0.2173 0.8577 0.2245
## [71,] 0.2141 0.8629 0.2271
## [72,] 0.2109 0.8635 0.2274
## [73,] 0.2058 0.8686 0.2314
## [74,] 0.2027 0.8755 0.2397
## [75,] 0.1997 0.8761 0.2404
## [76,] 0.1918 0.8830 0.2476
## [77,] 0.1889 0.8858 0.2512
## [78,] 0.1860 0.8870 0.2520
## [79,] 0.1814 0.8893 0.2556
## [80,] 0.1786 0.8950 0.2634
## [81,] 0.1758 0.8962 0.2635
```

We should choose a lower threshold (than 0.5) when the success criteria is very rare ie a very low percentage of the population will be positive

for churn data, delta is around 0.24

this lines up with the prop.table(table(Churned)) value where 0.217875 ppl churned (proportion of ‘yes’ in the data), so delta should be AROUND there (need not be exact)

The exact delta used is up to you, can depend on the users needs (if they want to prioritise TPR over FPR, etc)