Baruch College, STA-CIS 3920, Exercise#7 Anil Poonai 37

3 August 2020

**Exercise**

7.1

The sample function gave me the variables 3,8,4,7 in that order which means I will use the MedChg, SocEnr, Medicare, and SupSec variables from the hospital csv file. The code and its results is in the appendix at the bottom but I got a success rate of 70% and an error rate of 30%.

7.2

The code and results are in the appendix. I used the first three lagged and logged values for this regression and ended up with a success rate of 45.15% and an error rate of 54.85%. This does worst than both KNN and Naïve Bayes. Considerably worst as the values for those two were 55% and 52% success rate. It might be due to overfitting. It seems to be a common problem with regression models.

A picture containing circuit

Description automatically generated

This is the classification space using Lagged Logged 1 and 2 columns. I cannot see much here that is too useful as I cannot tell the difference even though there are two different colors here.

**Appendix**

R version 4.0.2 (2020-06-22) -- "Taking Off Again"

Copyright (C) 2020 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

[Previously saved workspace restored]

> library(tidyverse)

-- Attaching packages -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

v ggplot2 3.3.0 v purrr 0.3.4

v tibble 3.0.1 v dplyr 0.8.5

v tidyr 1.0.2 v stringr 1.4.0

v readr 1.3.1 v forcats 0.5.0

-- Conflicts ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --

x dplyr::filter() masks stats::filter()

x dplyr::lag() masks stats::lag()

> CIS\_STA\_3920\_LN7\_Hospital <- read\_csv("C:/Users/poona/Downloads/CIS-STA 3920 LN7 Hospital.csv")

Parsed with column specification:

cols(

State = col\_character(),

Phys = col\_double(),

Beds = col\_double(),

MedChg = col\_double(),

Medicare = col\_double(),

SocSec = col\_double(),

SocChg = col\_double(),

SupSec = col\_double(),

SocEnr = col\_double(),

Vote = col\_character()

)

> sample(1:8,4)

[1] 3 8 4 7

> glm.fit=glm(as.factor(Vote)~MedChg+SocEnr+Medicare+SupSec,data = CIS\_STA\_3920\_LN7\_Hospital,family = binomial)

> summary(glm.fit)

Call:

glm(formula = as.factor(Vote) ~ MedChg + SocEnr + Medicare +

SupSec, family = binomial, data = CIS\_STA\_3920\_LN7\_Hospital)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9595 -1.0973 0.5140 0.9372 1.7779

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.166e+00 3.794e+00 -0.571 0.5681

MedChg 3.016e-03 8.191e-02 0.037 0.9706

SocEnr 1.198e-06 5.454e-07 2.196 0.0281 \*

Medicare 2.579e-04 2.498e-04 1.032 0.3019

SupSec -1.129e-03 5.392e-04 -2.094 0.0363 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 68.593 on 49 degrees of freedom

Residual deviance: 58.295 on 45 degrees of freedom

AIC: 68.295

Number of Fisher Scoring iterations: 4

> glm.probs=predict(glm.fit,type = "response")

> glm.probs

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

0.29682168 0.13365858 0.63217494 0.31094167 0.91972968 0.52444778 0.67576130 0.52205072 0.97575521 0.43527427 0.44410635 0.45798817 0.77732768 0.74461170 0.75203293 0.63281998 0.13352406 0.18908114 0.51131380 0.52033974

21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

0.50622177 0.77822909 0.66804332 0.09349469 0.70739746 0.58575454 0.64389760 0.48103632 0.67097550 0.76259541 0.20588735 0.82379238 0.70097203 0.68484108 0.84212247 0.49973920 0.62869668 0.90700618 0.27943292 0.47909872

41 42 43 44 45 46 47 48 49 50

0.58246593 0.51929000 0.85336941 0.40744051 0.40301602 0.63672595 0.56748337 0.20089207 0.72291666 0.56740397

> glm.probs[glm.probs>.5]="Obama"

> glm.probs[glm.probs<.5]="McCain"

> glm.probs

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

"McCain" "McCain" "Obama" "McCain" "Obama" "Obama" "Obama" "Obama" "Obama" "McCain" "McCain" "McCain" "Obama" "Obama" "Obama" "Obama" "McCain" "McCain" "Obama" "Obama" "Obama" "Obama" "Obama" "McCain" "Obama"

26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

"Obama" "Obama" "McCain" "Obama" "Obama" "McCain" "Obama" "Obama" "Obama" "Obama" "McCain" "Obama" "Obama" "McCain" "McCain" "Obama" "Obama" "Obama" "McCain" "McCain" "Obama" "Obama" "McCain" "Obama" "Obama"

> table(glm.probs,CIS\_STA\_3920\_LN7\_Hospital$Vote)

glm.probs McCain Obama

McCain 12 5

Obama 10 23

> (12+23)/50

[1] 0.7

> AOS <- read\_csv("C:/Users/poona/Desktop/School/AOS.csv")

Parsed with column specification:

cols(

.default = col\_double(),

Date = col\_character(),

Direction = col\_character(),

HiLoRisk = col\_character()

)

See spec(...) for full column specifications.

> glm.fita=glm(as.factor(HiLoRisk)~LogR1+LogR2+LogR3,data = AOS,family = binomial)

> summary(glm.fita)

Call:

glm(formula = as.factor(HiLoRisk) ~ LogR1 + LogR2 + LogR3, family = binomial,

data = AOS)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.423 -1.172 -0.844 1.165 1.675

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.04985 0.03397 -1.467 0.142

LogR1 -129.75063 2188.52982 -0.059 0.953

LogR2 0.26401 0.06437 4.101 4.11e-05 \*\*\*

LogR3 129.96287 2188.52982 0.059 0.953

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 5057.1 on 3647 degrees of freedom

Residual deviance: 5019.3 on 3644 degrees of freedom

AIC: 5027.3

Number of Fisher Scoring iterations: 10

> glm.proba=predict(glm.fita,type = "response")

> glm.proba[1:50]

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

0.3928608 0.5188342 0.5174518 0.4964802 0.5067816 0.4983179 0.5459295 0.5887733 0.6180972 0.6054715 0.5686844 0.5304403 0.5149434 0.5119692 0.5009193 0.5207596 0.5423599 0.5380913 0.5529135 0.5971568 0.5545259 0.5116586

23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44

0.5020066 0.4833636 0.5411131 0.5835874 0.5637258 0.5092916 0.4947333 0.5094726 0.4977228 0.4790346 0.4667122 0.4956521 0.5293440 0.5288669 0.5247373 0.5572987 0.5553951 0.5287637 0.5198036 0.4818904 0.4624114 0.5031710

45 46 47 48 49 50

0.5131978 0.5379366 0.5551394 0.5141937 0.5037274 0.5140902

> glm.proba[glm.proba>=.5]="HiRisk"

> glm.proba[glm.proba<.5]="LoRisk"

> table(glm.proba,AOS$HiLoRisk)

glm.proba HiRisk LoRisk

HiRisk 851 1019

LoRisk 982 796

> (851+796)/3648

[1] 0.4514803

> AOS$HiLoRisk = ifelse(AOS$HiLoRisk=='HiRisk',1,0)

> glm.fit=glm(as.factor(HiLoRisk)~LogR1+LogR2,data = AOS,family = binomial)

> newdata=ProbeX

> X=AOS[,c(17,18)]

> StdX=apply(X,2,scale)

> dfX=as.data.frame(StdX)

> glm.probs=predict(glm.fit,newdata=dfX,type="response")

> StCard2=as.data.frame(cbind(dfX,AOS$HiLoRisk))

> glm.fit2=glm(AOS$HiLoRisk~LogR1+LogR2,data=StCard2,family=binomial)

> names(StProbeX)[1]="LogR1"

> names(StProbeX)[2]="LogR2"

> glm.probe=predict(glm.fit2,newdata=StProbeX,type="response")

> glm.y=glm.probe

> glm.y[glm.probe>.5]=1

> glm.y[glm.probe<.5]=0

> ProbeGlm(ProbeX=StProbeX,ProbeYhat = c(glm.y),InX = dfX,InY = AOS$HiLoRisk,xr=c(-3,3),yr=c(-3,3))

>

>

>

>