LogisticRegression.R

vijaykalmath

2022-01-04

```
# Logistic Regression in R

library(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Using Smarket Data

names(Smarket)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

```
# Direction is the target class

summary(Smarket)
```

```
##       Year           Lag1                Lag2                Lag3
##  Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000
##  1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000
##  Median :2003   Median : 0.039000   Median : 0.039000   Median : 0.038500
##  Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716
##  3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750
##  Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000
##       Lag4                Lag5              Volume           Today
##  Min.   :-4.922000   Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000
##  1st Qu.:-0.640000   1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500
##  Median : 0.038500   Median : 0.03850   Median :1.4229   Median : 0.038500
##  Mean   : 0.001636   Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138
##  3rd Qu.: 0.596750   3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750
##  Max.   : 5.733000   Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000
##  Direction
##  Down:602
##  Up  :648
##
##
##
##
```

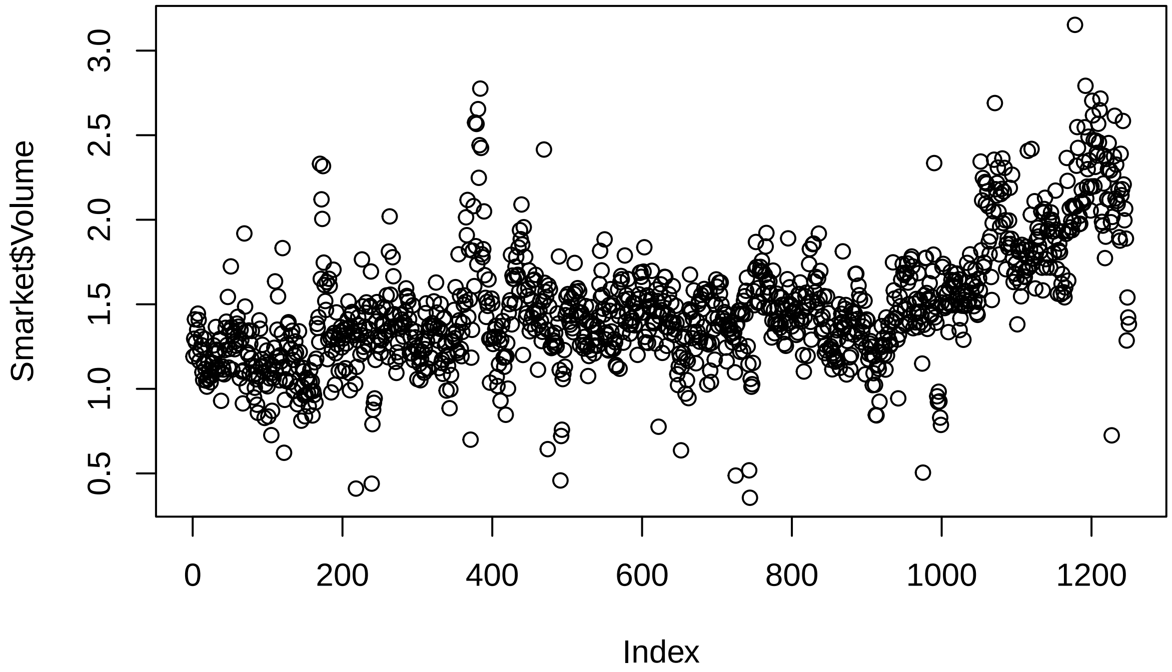
```
# Correlation between columns

cor(Smarket[,-9])
```

```
##              Year         Lag1         Lag2         Lag3         Lag4
## Year   1.00000000  0.029699649  0.030596422  0.033194581  0.035688718
## Lag1   0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911
## Lag2   0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533
## Lag3   0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036
## Lag4   0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000
## Lag5   0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641
## Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246
## Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527
##                Lag5      Volume        Today
## Year    0.029787995  0.53900647  0.030095229
## Lag1   -0.005674606  0.04090991 -0.026155045
## Lag2   -0.003557949 -0.04338321 -0.010250033
## Lag3   -0.018808338 -0.04182369 -0.002447647
## Lag4   -0.027083641 -0.04841425 -0.006899527
## Lag5    1.000000000 -0.02200231 -0.034860083
## Volume -0.022002315  1.00000000  0.014591823
## Today  -0.034860083  0.01459182  1.000000000
```

```
# We can see that Volume and Year have high correlation

plot(Smarket$Volume)
```



```
# Lets fit Logistic Regression Model using glm func


glm.fit = glm(Direction ~ . -Today -Year ,data = Smarket,family = binomial )

summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ . - Today - Year, family = binomial,
##     data = Smarket)
##
## Deviance Residuals:
##    Min       1Q   Median       3Q      Max
## -1.446   -1.203   1.065    1.145    1.326
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1741.6
##
## Number of Fisher Scoring iterations: 3
```

```
# Using predict function

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

glm.probs[1:10]
```

```
##         1         2         3         4         5         6         7         8
## 0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 0.5092292
##         9        10
## 0.5176135 0.4888378
```

```
glm.predict = ifelse(glm.probs > 0.5 , "Up","Down")

# Create Table between glm.predict and Direction from Smarket

table(glm.predict,Smarket$Direction)
```

```
##
## glm.predict Down  Up
##        Down  145 141
##        Up    457 507
```

```
mean(glm.predict==Smarket$Direction)
```

```
## [1] 0.5216
```

```
# Lets train glm model with only a subset now .

# Subset Condition = Year < 2005

subset_condition  = (Smarket$Year < 2005 )

Smarket_2005 = Smarket[!subset_condition,]

newglm.fit = glm(Direction ~ . -Today -Year ,data = Smarket,family = binomial,subset = subset_condition


summary(newglm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ . - Today - Year, family = binomial,
##     data = Smarket, subset = subset_condition)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.302  -1.190   1.079   1.160   1.350
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.191213   0.333690   0.573    0.567
## Lag1        -0.054178   0.051785  -1.046    0.295
## Lag2        -0.045805   0.051797  -0.884    0.377
## Lag3         0.007200   0.051644   0.139    0.889
## Lag4         0.006441   0.051706   0.125    0.901
## Lag5        -0.004223   0.051138  -0.083    0.934
## Volume      -0.116257   0.239618  -0.485    0.628
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1383.3  on 997  degrees of freedom
## Residual deviance: 1381.1  on 991  degrees of freedom
## AIC: 1395.1
##
## Number of Fisher Scoring iterations: 3
```

```
newglm.probs = predict(newglm.fit,Smarket_2005,type="response")

newglm.probs[1:10]
```

```
##       999      1000      1001      1002      1003      1004      1005      1006
## 0.5282195 0.5156688 0.5226521 0.5138543 0.4983345 0.5010912 0.5027703 0.5095680
##      1007      1008
## 0.5040112 0.5106408
```

```
newglm.predict = ifelse(newglm.probs > 0.5 , "Up","Down")

# Create Table between glm.predict and Direction from Smarket

table(newglm.predict,Smarket_2005$Direction)
```

```
##
## newglm.predict Down Up
##           Down   77 97
##           Up     34 44
```

```
mean(newglm.predict==Smarket_2005$Direction)
```

```
## [1] 0.4801587
```

```
mean(newglm.predict!=Smarket_2005$Direction)
```

```
## [1] 0.5198413
```

```
# Testing with only few columns

customglm.fit = glm(Direction ~ Lag1 + Lag2 + Lag1:Lag2 ,data = Smarket,family = binomial,subset = subs


summary(customglm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag1:Lag2, family = binomial,
##     data = Smarket, subset = subset_condition)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.334  -1.189   1.077   1.163   1.338
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.03214    0.06339   0.507    0.612
## Lag1        -0.05603    0.05213  -1.075    0.283
## Lag2        -0.04455    0.05167  -0.862    0.389
## Lag1:Lag2   -0.00208    0.03411  -0.061    0.951
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1383.3  on 997  degrees of freedom
## Residual deviance: 1381.4  on 994  degrees of freedom
## AIC: 1389.4
##
## Number of Fisher Scoring iterations: 3
```

```
customglm.probs = predict(customglm.fit,Smarket_2005,type="response")

customglm.probs[1:10]
```

```
##       999      1000      1001      1002      1003      1004      1005      1006
## 0.5098227 0.5208330 0.5328859 0.5258757 0.5072284 0.5061546 0.5048635 0.5128758
##      1007      1008
## 0.5093808 0.5158634
```

```
customglm.predict = ifelse(customglm.probs > 0.5 , "Up","Down")

# Create Table between glm.predict and Direction from Smarket

table(customglm.predict,Smarket_2005$Direction)
```

```
##
## customglm.predict Down  Up
##              Down   35  35
##              Up     76 106
```

```
mean(customglm.predict==Smarket_2005$Direction)
```

```
## [1] 0.5595238
```

```
mean(customglm.predict!=Smarket_2005$Direction)
```

```
## [1] 0.4404762
```

```
customglm.fit = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag1:Lag2 + Lag1:Lag3 + Lag2:Lag3 + Lag1:Lag2:Lag3


summary(customglm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag1:Lag2 + Lag1:Lag3 +
##     Lag2:Lag3 + Lag1:Lag2:Lag3, family = binomial, data = Smarket,
##     subset = subset_condition)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.758  -1.191   1.004   1.160   1.498
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.033707   0.063507   0.531    0.596
## Lag1           -0.044698   0.053154  -0.841    0.400
## Lag2           -0.045539   0.052446  -0.868    0.385
## Lag3            0.002368   0.052232   0.045    0.964
## Lag1:Lag2      -0.018539   0.037493  -0.494    0.621
## Lag1:Lag3       0.036703   0.032382   1.133    0.257
## Lag2:Lag3       0.017313   0.035258   0.491    0.623
## Lag1:Lag2:Lag3 -0.020089   0.019934  -1.008    0.314
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1383.3  on 997  degrees of freedom
## Residual deviance: 1378.4  on 990  degrees of freedom
## AIC: 1394.4
##
## Number of Fisher Scoring iterations: 4
```

```
customglm.probs = predict(customglm.fit,Smarket_2005,type="response")

customglm.probs[1:10]
```

```
##       999      1000      1001      1002      1003      1004      1005      1006
## 0.5098410 0.5184495 0.5287525 0.5318157 0.5058649 0.5058795 0.5076375 0.5126681
##      1007      1008
## 0.5130118 0.5174902
```

```
customglm.predict = ifelse(customglm.probs > 0.5 , "Up","Down")

# Create Table between glm.predict and Direction from Smarket

table(customglm.predict,Smarket_2005$Direction)
```

```
##
## customglm.predict Down  Up
##              Down   31  22
##              Up     80 119
```

```
mean(customglm.predict==Smarket_2005$Direction)
```

```
## [1] 0.5952381
```

```
mean(customglm.predict!=Smarket_2005$Direction)
```

```
## [1] 0.4047619
```