DaigleInClassLabWk12D3.R

2011home

Sun Apr 15 12:49:18 2018

```
# Chris Daigle
# Week12 Day 3 - 13 April

# Exercise

setwd(
  "/Users/2011home/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/R/DataSets"
)
# Regression model: examine the relation between X and Y
# Y|X ~ P(\beta)
# E(Y|X) = beta0 + beta1 * X
# Y = beta0 + beta1 X + u,  E(u|X)=0,
# In this mean regression, we may assess model validity: Estimation and
# Inference on beta
# We may predict Y using X: prediction

# College data: Demographic characteristics, tuition, and more for USA colleges.

# Private: Public/private indicator
# Apps: Number of applications received
# Accept: Number of applicants accepted
# Enroll: Number of new students enrolled
# Top10perc: New students from top 10 % of high school class
# Top25perc: New students from top 25 % of high school class
# F.Undergrad: Number of full-time undergraduates
# P.Undergrad: Number of part-time undergraduates
# Outstate: Out-of-state tuition
# Room.Board: Room and board costs
# Books: Estimated book costs
# Personal: Estimated personal spending
# PhD: Percent of faculty with Ph.D.'s
# Terminal: Percent of faculty with terminal degree
# S.F.Ratio: Student/faculty ratio
# perc.alumni: Percent of alumni who donate
# Expend: Instructional expenditure per student
# Grad.Rate: Graduation rate

college <- read.csv("College.csv")
college1 <- college[, c(-1, -7, -8, -9, -10, -11, -12, -14, -15)]

# You can create a dummy variable from a continuous variable. For example,
college1$dummyPC <-
  as.numeric(college1$perc.alumni > mean(college1$perc.alumni))

# Find the variables that provide good prediction for this dummy varaible.

naiveReg <- lm(dummyPC ~ ., data = college1)
```

```r
summary(naiveReg)
```

```
##
## Call:
## lm(formula = dummyPC ~ ., data = college1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76584 -0.21996 -0.01346  0.20028  0.70500
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.518e-01  8.881e-02  -3.962 8.14e-05 ***
## PrivateYes   4.132e-02  3.292e-02   1.255   0.2098
## Apps        -1.220e-05  9.429e-06  -1.293   0.1963
## Accept       7.196e-06  1.792e-05   0.402   0.6881
## Enroll       4.809e-06  3.008e-05   0.160   0.8730
## Top10perc   -1.355e-03  9.004e-04  -1.505   0.1327
## Personal    -2.918e-06  1.650e-05  -0.177   0.8597
## S.F.Ratio    1.573e-03  3.498e-03   0.450   0.6531
## perc.alumni  3.263e-02  1.067e-03  30.595  < 2e-16 ***
## Expend      -1.076e-07  3.051e-06  -0.035   0.9719
## Grad.Rate    1.301e-03  7.636e-04   1.704   0.0887 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2816 on 766 degrees of freedom
## Multiple R-squared:  0.686,  Adjusted R-squared:  0.6819
## F-statistic: 167.4 on 10 and 766 DF,  p-value: < 2.2e-16
```

```r
# This makes sense that the variable that contructed the new variable, dummyPC
# and perc.alumni, would have a statistically significant relationshop So, let's
# be a little more sophisticated, but nott too much - stick with a linear model
# and first remove the variable that, without a doubt, constructs the variable
# of interest (per.alumni)
college2 <- college1[, -8]
head(college2)
```

```
##   Private Apps Accept Enroll Top10perc Personal S.F.Ratio Expend Grad.Rate
## 1     Yes 1660   1232    721        23     2200      18.1   7041        60
## 2     Yes 2186   1924    512        16     1500      12.2  10527        56
## 3     Yes 1428   1097    336        22     1165      12.9   8735        54
## 4     Yes  417    349    137        60      875       7.7  19016        59
## 5     Yes  193    146     55        16     1500      11.9  10922        15
## 6     Yes  587    479    158        38      675       9.4   9727        55
##   dummyPC
## 1       0
## 2       0
## 3       1
## 4       1
## 5       0
## 6       0
```

```r
reg1 <- lm(dummyPC ~ ., data = college2)
summary(reg1)
```

```
##
## Call:
## lm(formula = dummyPC ~ ., data = college2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1676 -0.3692 -0.0282  0.3425  1.0094
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.973e-02  1.317e-01  -0.681 0.495807
## PrivateYes   1.733e-01  4.862e-02   3.565 0.000386 ***
## Apps        -3.345e-05  1.401e-05  -2.388 0.017177 *
## Accept      -5.486e-06  2.669e-05  -0.206 0.837214
## Enroll       7.081e-05  4.470e-05   1.584 0.113560
## Top10perc    4.221e-03  1.314e-03   3.213 0.001367 **
## Personal    -7.318e-05  2.434e-05  -3.007 0.002724 **
## S.F.Ratio   -6.951e-03  5.195e-03  -1.338 0.181233
## Expend       8.445e-06  4.526e-06   1.866 0.062430 .
## Grad.Rate    7.413e-03  1.098e-03   6.752 2.89e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4194 on 767 degrees of freedom
## Multiple R-squared:  0.3023, Adjusted R-squared:  0.2942
## F-statistic: 36.93 on 9 and 767 DF,  p-value: < 2.2e-16
```

```r
# Signigicant: PrivateYes, Apps, Top10Perc, Personal, Grad.Rate Let's increase
# the degree of freedom by removing variables that are deemed insignificant
head(college2)
```

```
##   Private Apps Accept Enroll Top10perc Personal S.F.Ratio Expend Grad.Rate
## 1     Yes 1660   1232    721        23     2200      18.1   7041        60
## 2     Yes 2186   1924    512        16     1500      12.2  10527        56
## 3     Yes 1428   1097    336        22     1165      12.9   8735        54
## 4     Yes  417    349    137        60      875       7.7  19016        59
## 5     Yes  193    146     55        16     1500      11.9  10922        15
## 6     Yes  587    479    158        38      675       9.4   9727        55
##   dummyPC
## 1       0
## 2       0
## 3       1
## 4       1
## 5       0
## 6       0
```

```r
reg2 <-
  lm(dummyPC ~ Private + Apps + Top10perc + Personal + Grad.Rate, data = college2)
summary(reg2)
```

```
##
## Call:
## lm(formula = dummyPC ~ Private + Apps + Top10perc + Personal +
##     Grad.Rate, data = college2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.06051 -0.36616 -0.01369  0.34794  1.11425
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.682e-01  7.851e-02  -2.142  0.03249 *
## PrivateYes   1.926e-01  4.333e-02   4.445 1.01e-05 ***
## Apps        -2.229e-05  4.984e-06  -4.472 8.92e-06 ***
## Top10perc    6.162e-03  1.058e-03   5.824 8.46e-09 ***
## Personal    -6.519e-05  2.408e-05  -2.707  0.00694 **
## Grad.Rate    7.329e-03  1.099e-03   6.669 4.91e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4215 on 771 degrees of freedom
## Multiple R-squared:  0.2918, Adjusted R-squared:  0.2872
## F-statistic: 63.52 on 5 and 771 DF,  p-value: < 2.2e-16
```

```r
# All but Personal and the intercept exude statistical significance of 99.99%
# level
#
# Let's step it up a notch and use the logistic model as the variable of
# interest is not continuous. A non-linear model is more appropiate.
logistic1 <- glm((dummyPC == 1) ~ ., data=college2, family="binomial")
summary(logistic1)
```

```
##
## Call:
## glm(formula = (dummyPC == 1) ~ ., family = "binomial", data = college2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9862  -0.8509  -0.2803   0.8343   2.3641
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.828e+00  8.238e-01  -4.647 3.37e-06 ***
## PrivateYes   8.928e-01  3.068e-01   2.910  0.00362 **
## Apps        -2.928e-04  1.072e-04  -2.731  0.00631 **
## Accept      -2.312e-04  1.952e-04  -1.185  0.23618
## Enroll       1.081e-03  3.357e-04   3.220  0.00128 **
## Top10perc    2.531e-02  8.114e-03   3.119  0.00181 **
## Personal    -4.091e-04  1.412e-04  -2.898  0.00376 **
## S.F.Ratio   -2.777e-02  3.135e-02  -0.886  0.37585
## Expend       9.970e-05  3.781e-05   2.637  0.00837 **
## Grad.Rate    4.165e-02  6.547e-03   6.362 2.00e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1073.80  on 776  degrees of freedom
## Residual deviance:  785.46  on 767  degrees of freedom
## AIC: 805.46
```

```
##
## Number of Fisher Scoring iterations: 5
```

```r
# Significant: PrivateYes, Apps, Enroll, Top10perc, Personal, Expend, Grad.Rate
head(college2)
```

```
##   Private Apps Accept Enroll Top10perc Personal S.F.Ratio Expend Grad.Rate
## 1     Yes 1660   1232    721        23     2200      18.1   7041        60
## 2     Yes 2186   1924    512        16     1500      12.2  10527        56
## 3     Yes 1428   1097    336        22     1165      12.9   8735        54
## 4     Yes  417    349    137        60      875       7.7  19016        59
## 5     Yes  193    146     55        16     1500      11.9  10922        15
## 6     Yes  587    479    158        38      675       9.4   9727        55
##   dummyPC
## 1       0
## 2       0
## 3       1
## 4       1
## 5       0
## 6       0
```

```r
college3 <- college2[, c(-3, -7)]
logistic2 <- glm((dummyPC == 1) ~ ., data=college3, family="binomial")
summary(logistic2)
```

```
##
## Call:
## glm(formula = (dummyPC == 1) ~ ., family = "binomial", data = college3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9173  -0.8419  -0.2867   0.8317   2.3899
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.376e+00  5.344e-01  -8.189 2.64e-16 ***
## PrivateYes   8.838e-01  2.975e-01   2.971 0.002969 **
## Apps        -3.837e-04  7.659e-05  -5.010 5.46e-07 ***
## Enroll       8.199e-04  2.606e-04   3.147 0.001652 **
## Top10perc    2.793e-02  7.851e-03   3.558 0.000374 ***
## Personal    -3.949e-04  1.401e-04  -2.817 0.004841 **
## Expend       1.141e-04  3.387e-05   3.369 0.000754 ***
## Grad.Rate    4.096e-02  6.501e-03   6.301 2.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1073.80  on 776  degrees of freedom
## Residual deviance:  787.73  on 769  degrees of freedom
## AIC: 803.73
##
## Number of Fisher Scoring iterations: 5
```

```r
# Variables that are likely to influence investment above the average percent of
# alumni who donate: Private, Enroll, Top10perc, Expend, and Grad.Rate Variables
```

```r
# that are likely to influence investment below the average percent of alumni
# who donate: Apps and Personal
pHat <- fitted(logistic2)
yHat <- round(pHat)
table(college2$dummyPC)
```

```
##
##   0   1
## 414 363
```

```r
table(yHat, y.true=college2$dummyPC)
```

```
##     y.true
## yHat   0   1
##    0 314  91
##    1 100 272
```

```r
100/414
```

```
## [1] 0.2415459
```

```r
272/363
```

```
## [1] 0.7493113
```

```r
# This model seems prerry bad at predicting if there will not be a greater than
# average proportion of alumni investing, but not too bad at predicting if a
# proportion above average of alumni will invest. The model correctly predicts
# about above average percent of alumni investing at about 75%. OR! Incorrectly
# does so about 1/4 of the time. The model correctly predicts below average
# percent of alumni investing at about 25%. OR! Incorrectly does so about 3/4 of
# the time.

# # Let's see if we can get it better by applying some weights. There are 8
# # variables if we account for the intercept, so let's make a vector of 7 values
# # for probability weights with some intuition - base them on the P-values.
# summary(logistic2)
# lambda <- c(2, 3, 2, 3, 2, 3, 3)
# x <- 2 + 3 + 2 + 3 + 2 + 3 + 3
# lambda <- 1 / x * lambda
# lambda
# logistic3 <-
#   glm((dummyPC == 1) ~ .,
#       data = college3,
#       weights = c(3 / x, 2 / x, 3 / x, 2 / x, 3 / x, 3 / x),
#       family = "binomial"
#   )
# This is not the right way to use weights, I think I need a vector as long as
# the number of observations, not as long as the variables (which seems weird to
# me), to apply the weight command in the glm regression.
# summary(logistic3)
#
# pHat1 <- fitted(logistic3)
# yHat1 <- round(pHat1)
# table(college2$dummyPC)
# table(yHat1, y.true = college2$dummyPC)
```