ISLR: Classification

Justin M Shea

Classification

Logistic Regression

Example on `Default` credit card data

```
library(ISLR)
str(Default)
```

```
## 'data.frame':    10000 obs. of  4 variables:
##  $ default: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ student: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ balance: num  730 817 1074 529 786 ...
##  $ income : num  44362 12106 31767 35704 38463 ...
```

```
?Default
```

```
balance_default <- glm(default ~ balance, data = Default, family="binomial")
summary(balance_default)
```

```
##
## Call:
## glm(formula = default ~ balance, family = "binomial", data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <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: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

```
student_default <- glm(default ~ student, data = Default, family="binomial")
summary(student_default)
```

```
##
## Call:
## glm(formula = default ~ student, family = "binomial", data = Default)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2970  -0.2970  -0.2434  -0.2434   2.6585
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
## studentYes   0.40489    0.11502    3.52 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2908.7  on 9998  degrees of freedom
## AIC: 2912.7
##
## Number of Fisher Scoring iterations: 6
```

Example on `Smarket` stock market data

```
?Smarket
```

```
str(Smarket)
```

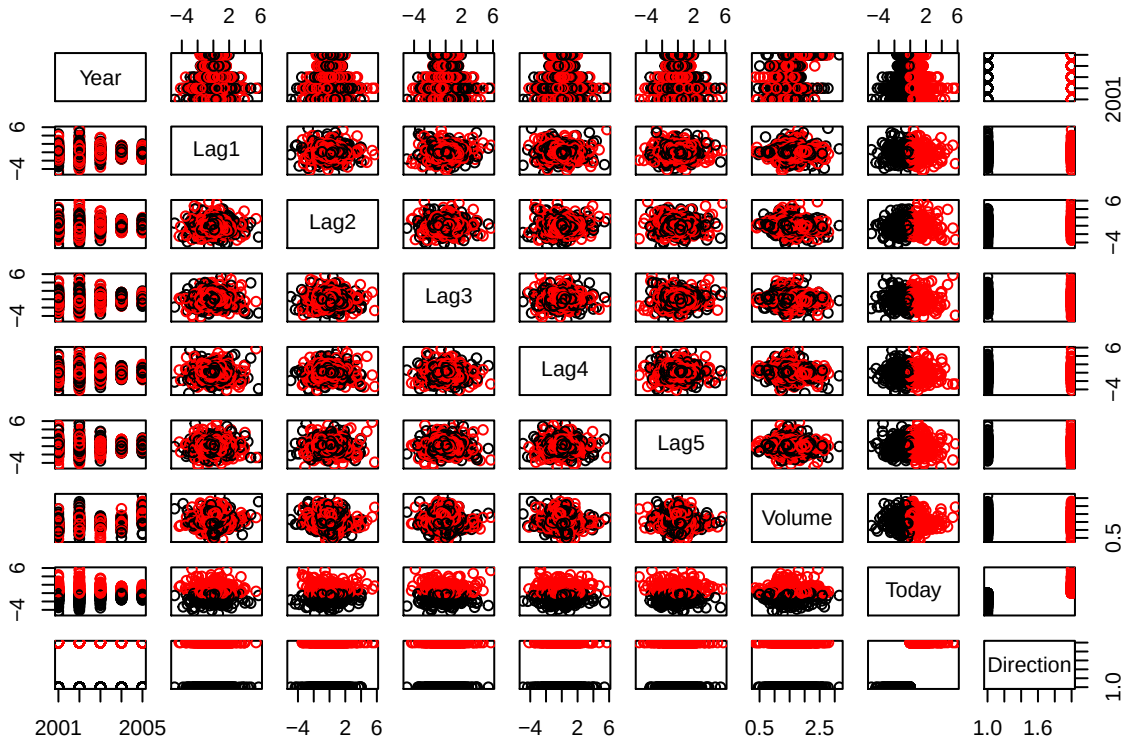
```
## 'data.frame':    1250 obs. of  9 variables:
##  $ Year     : num  2001 2001 2001 2001 2001 ...
##  $ Lag1     : num  0.381 0.959 1.032 -0.623 0.614 ...
##  $ Lag2     : num  -0.192 0.381 0.959 1.032 -0.623 ...
##  $ Lag3     : num  -2.624 -0.192 0.381 0.959 1.032 ...
##  $ Lag4     : num  -1.055 -2.624 -0.192 0.381 0.959 ...
##  $ Lag5     : num  5.01 -1.055 -2.624 -0.192 0.381 ...
##  $ Volume   : num  1.19 1.3 1.41 1.28 1.21 ...
##  $ Today    : num  0.959 1.032 -0.623 0.614 0.213 ...
##  $ Direction: Factor w/ 2 levels "Down","Up": 2 2 1 2 2 2 1 2 2 2 ...
```

```
summary(Smarket)
```

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

```
##  Mean   :1.4783   Mean   : 0.003138
##  3rd Qu.:1.6417   3rd Qu.: 0.596750
##  Max.   :3.1525   Max.   : 5.733000
```

```
pairs(Smarket, col=Smarket$Direction)
```




```
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
               data = Smarket, family = binomial)

summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, 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
```

```
glm.probs <- predict(glm.fit, type = "response")
glm.probs[1:5]
```

```
##         1         2         3         4         5
## 0.5070841 0.4814679 0.4811388 0.5152224 0.5107812
```

```
glm.pred <- ifelse(glm.probs > 0.5, "Up", "Down")

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

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

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

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

Make training and test set

```
  train <- Smarket$Year < 2005

glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
               data=Smarket, family=binomial, subset=train)

     glm.probs <- predict(glm.fit, newdata=Smarket[!train, ], type = "response")
     glm.pred  <- ifelse(glm.probs > 0.5, "Up", "Down")
Direction.2005 <- Smarket$Direction[!train]
```

```
table(glm.pred, Direction.2005)
```

```
##         Direction.2005
## glm.pred Down Up
##     Down   77 97
##     Up     34 44
```

```
mean(glm.pred==Direction.2005)
```

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

Fit smaller model

```
  glm.fit <- glm(Direction~Lag1+Lag2, data=Smarket,
                 family=binomial, subset=train)
```

```
glm.probs <- predict(glm.fit, newdata=Smarket[!train, ], type = "response")
 glm.pred <- ifelse(glm.probs > 0.5, "Up", "Down")
```

```
table(glm.pred, Direction.2005)
```

```
##         Direction.2005
## glm.pred Down  Up
##     Down   35  35
##     Up     76 106
```

```
mean(glm.pred==Direction.2005)
```

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

```
106/(76+106)
```

```
## [1] 0.5824176
```

```
predict(glm.fit,newdata=data.frame(Lag1=c(1.2,1.5),Lag2=c(1.1,-0.8)),type="response")
```

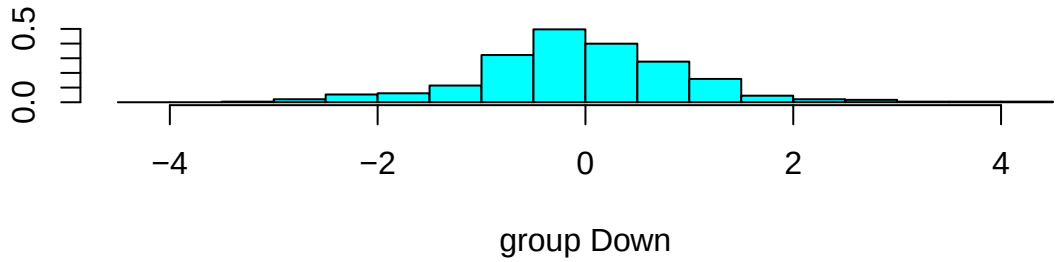
```
##         1         2
## 0.4791462 0.4960939
```

Linear Discriminant Analysis

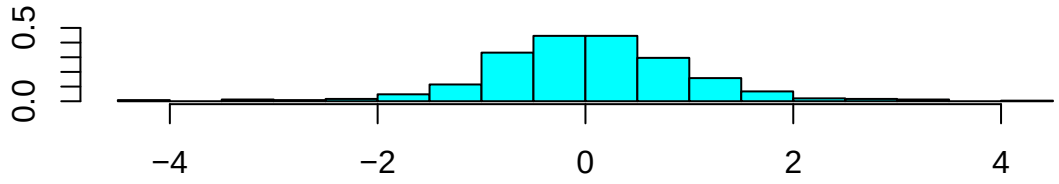
```
library(MASS)
lda.fit <- lda(Direction~Lag1+Lag2, data=Smarket, subset = Year<2005)
lda.fit
```

```
## Call:
## lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = Year <
##     2005)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
##
## Coefficients of linear discriminants:
##             LD1
## Lag1 -0.6420190
## Lag2 -0.5135293
```

```
plot(lda.fit)
```



group Down



group Up

```
Smarket.2005 <- subset(Smarket, Year==2005)
lda.pred <- predict(lda.fit, Smarket.2005)
class(lda.pred)
```

```
## [1] "list"
```

```
data.frame(lda.pred)[1:5,]
```

```
##      class posterior.Down posterior.Up         LD1
## 999     Up      0.4901792    0.5098208  0.08293096
## 1000    Up      0.4792185    0.5207815  0.59114102
## 1001    Up      0.4668185    0.5331815  1.16723063
## 1002    Up      0.4740011    0.5259989  0.83335022
## 1003    Up      0.4927877    0.5072123 -0.03792892
```

```
table(lda.pred$class, Smarket.2005$Direction)
```

```
##
##        Down  Up
##   Down   35  35
##   Up     76 106
```

```
mean(lda.pred$class==Smarket.2005$Direction)
```

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

Quadratic Discriminant Analysis

```
qda.fit <- qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
qda.fit
```

```
## Call:
## qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
##
## Prior probabilities of groups:
##     Down       Up
## 0.491984 0.508016
##
## Group means:
##             Lag1        Lag2
## Down  0.04279022  0.03389409
## Up   -0.03954635 -0.03132544
```

```
qda.class <- predict(qda.fit, Smarket.2005)$class

table(qda.class, Direction.2005)
```

```
##          Direction.2005
## qda.class Down  Up
##      Down   30  20
##      Up     81 121
```

```
mean(qda.class==Direction.2005)
```

```
## [1] 0.5992063
```

K-Nearest Neighbors

```
library(class)
```

```
?knn
```

```
attach(Smarket)
Xlag <- cbind(Lag1,Lag2)
train <- Year<2005
knn.pred <- knn(Xlag[train,], Xlag[!train,], Direction[train], k=1)
```

```
table(knn.pred, Direction[!train])
```

```
##
## knn.pred Down Up
##     Down   43 58
##     Up     68 83
```

```
mean(knn.pred==Direction[!train])
```

```
## [1] 0.5
```