Math642\_HW6\_FyonaSun

Fyona Sun

2/24/2020

## 4.1

Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representation and logit representation for the logistic regression model are equivalent.

## 4.6

Suppose we collect data for a group of students in a statistics class with variables X1 = hours studied, X2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, , ,

1. Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

exp(-6+0.05\*40+3.5)/(1+exp(-6+0.05\*40+3.5))

## [1] 0.3775407

1. How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?
2. give X2=3.5, solve for X1

$$e^{-6+0.05X\_1+3.6}=1\\
X1 = \frac{log(1)+6-3.5}{0.05}=50$$

(log(1)+6-3.5)/0.05

## [1] 50

## 4.9

This problem has to do with odds. (a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

$$\frac{p(X)}{1-P(x)}=0.37 \\
p(X) = \frac{0.37}{1+0.37}=0.27$$

Thus on average there are 27% of people defaulting on their creditt card payment.

1. Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default?

The odds that she will default is 19%.

## 4.10

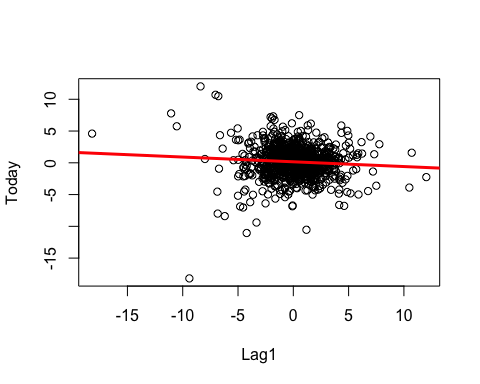
This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter’s lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

1. Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

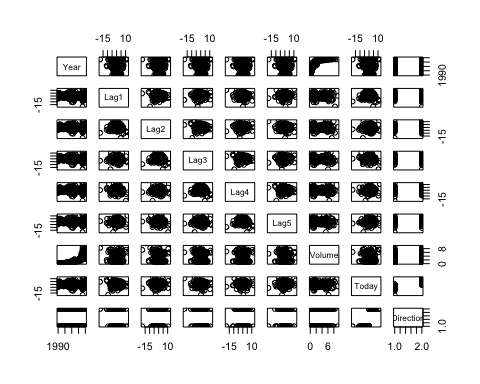
library(ISLR)  
attach(Weekly)  
summary(Weekly)

## Year Lag1 Lag2 Lag3   
## Min. :1990 Min. :-18.1950 Min. :-18.1950 Min. :-18.1950   
## 1st Qu.:1995 1st Qu.: -1.1540 1st Qu.: -1.1540 1st Qu.: -1.1580   
## Median :2000 Median : 0.2410 Median : 0.2410 Median : 0.2410   
## Mean :2000 Mean : 0.1506 Mean : 0.1511 Mean : 0.1472   
## 3rd Qu.:2005 3rd Qu.: 1.4050 3rd Qu.: 1.4090 3rd Qu.: 1.4090   
## Max. :2010 Max. : 12.0260 Max. : 12.0260 Max. : 12.0260   
## Lag4 Lag5 Volume   
## Min. :-18.1950 Min. :-18.1950 Min. :0.08747   
## 1st Qu.: -1.1580 1st Qu.: -1.1660 1st Qu.:0.33202   
## Median : 0.2380 Median : 0.2340 Median :1.00268   
## Mean : 0.1458 Mean : 0.1399 Mean :1.57462   
## 3rd Qu.: 1.4090 3rd Qu.: 1.4050 3rd Qu.:2.05373   
## Max. : 12.0260 Max. : 12.0260 Max. :9.32821   
## Today Direction   
## Min. :-18.1950 Down:484   
## 1st Qu.: -1.1540 Up :605   
## Median : 0.2410   
## Mean : 0.1499   
## 3rd Qu.: 1.4050   
## Max. : 12.0260

plot(Today~Lag1, data=Weekly)  
simplelm = lm(Today~Lag1, data=Weekly)  
abline(simplelm, lwd= 3, col= "red")



pairs(Weekly)

 (b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

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

##   
## Call:  
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +   
## Volume, family = binomial, data = Weekly)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6949 -1.2565 0.9913 1.0849 1.4579   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.26686 0.08593 3.106 0.0019 \*\*  
## Lag1 -0.04127 0.02641 -1.563 0.1181   
## Lag2 0.05844 0.02686 2.175 0.0296 \*   
## Lag3 -0.01606 0.02666 -0.602 0.5469   
## Lag4 -0.02779 0.02646 -1.050 0.2937   
## Lag5 -0.01447 0.02638 -0.549 0.5833   
## Volume -0.02274 0.03690 -0.616 0.5377   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1496.2 on 1088 degrees of freedom  
## Residual deviance: 1486.4 on 1082 degrees of freedom  
## AIC: 1500.4  
##   
## Number of Fisher Scoring iterations: 4

The variable lag2 has a p-value of 0.0296 < 0.05 which is statistically significant.

1. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

probs<- predict(fit, type='response')  
pred<- rep('Down',length(probs))  
pred[probs>0.5] <- 'Up'  
table(pred, Direction)

## Direction  
## pred Down Up  
## Down 54 48  
## Up 430 557

cm<- table(pred, Direction)  
 TP<-cm[2,2]  
 TN<-cm[1,1]  
 FP<-cm[2,1]  
 FN<-cm[1,2]  
 N<-sum(cm)  
 acc<-(TP+TN)/N  
 sens<-TP/(TP+FN)  
 prec<-TP/(TP+FP)  
 FPR<-FP/(FP+TN)  
 spec<-1-FPR  
   
 acc

## [1] 0.5610652

sens

## [1] 0.9206612

prec

## [1] 0.5643364

FPR

## [1] 0.8884298

spec

## [1] 0.1115702

The accuracy rate is 56.10652%. The true positive rate is 92.06612% which also known as the sensiticity. The false positive rate is 88.84298%, which equals to 1-specificity.

1. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

train.data = Weekly[Weekly$Year<2009,]  
test.data = Weekly[Weekly$Year>2008,]  
fit2 = glm(Direction~Lag2, data= train.data, family = "binomial")  
summary(fit2)

##   
## Call:  
## glm(formula = Direction ~ Lag2, family = "binomial", data = train.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.536 -1.264 1.021 1.091 1.368   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.20326 0.06428 3.162 0.00157 \*\*  
## Lag2 0.05810 0.02870 2.024 0.04298 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1354.7 on 984 degrees of freedom  
## Residual deviance: 1350.5 on 983 degrees of freedom  
## AIC: 1354.5  
##   
## Number of Fisher Scoring iterations: 4

probs = predict(fit2, type="response", newdata = test.data)  
testdirs = Weekly$Direction[Weekly$Year>2008]  
pred<- rep('Down',length(probs))  
pred[probs>0.5] <- 'Up'  
table(pred, test.data$Direction)

##   
## pred Down Up  
## Down 9 5  
## Up 34 56

#the overall fraction of correct predictions for the held out data  
mean(pred==test.data$Direction)

## [1] 0.625

The accuracy is 62.5% (e) Repeat (d) using LDA.

require(MASS)

## Loading required package: MASS

fit.lda <- lda(Direction ~ Lag2, data = train.data)  
  
pred <- predict(fit.lda, newdata = test.data)  
pred\_values <- pred$class  
table(pred\_values, test.data$Direction)

##   
## pred\_values Down Up  
## Down 9 5  
## Up 34 56

acc <- paste('Accuracy:', mean(pred\_values == test.data$Direction))  
acc

## [1] "Accuracy: 0.625"

The LDA model also gives 62.5% (f) Repeat (d) using QDA.

fit.qda <- qda(Direction ~ Lag2, data = train.data)  
  
pred <- predict(fit.qda, newdata = test.data)  
pred\_values <- pred$class  
table(pred\_values, test.data$Direction)

##   
## pred\_values Down Up  
## Down 0 0  
## Up 43 61

acc <- paste('Accuracy:', mean(pred\_values == test.data$Direction))  
acc

## [1] "Accuracy: 0.586538461538462"

The QDA model does not do as well as the LDA model. The accuracy of this model is 58.65%. (g) Repeat (d) using KNN with K = 1.

require(class)

## Loading required package: class

knn\_pred <- knn(train = data.frame(train.data$Lag2),   
 test = data.frame(test.data$Lag2),   
 cl = train.data$Direction, k = 1)  
  
acc <- paste('Accuracy:', mean(knn\_pred == test.data$Direction))  
acc

## [1] "Accuracy: 0.5"

The KNN model with k=1 gives a model with accuracy = 0.5

1. Which of these methods appears to provide the best results on this data? The LDA model with Lag2 seems to provide the best result.
2. Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confu- sion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.2

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

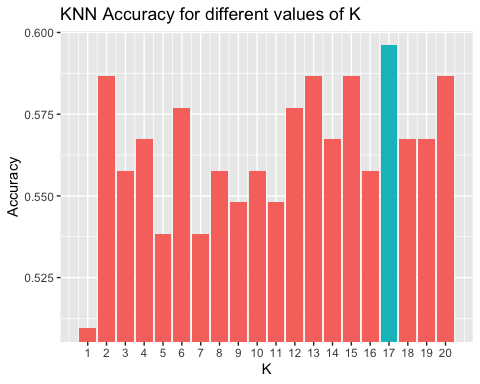
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.2

acc <- list('1' = 0.5)  
for (i in 1:20) {  
 knn\_pred <- knn(train = data.frame(train.data$Lag2), test = data.frame(test.data$Lag2), cl = train.data$Direction, k = i)  
 acc[as.character(i)] = mean(knn\_pred == test.data$Direction)  
}  
  
acc <- unlist(acc)  
data.frame(acc = acc)%>%  
 mutate(k = row\_number())%>%  
 ggplot(aes(k, acc)) +  
 geom\_col(aes(fill = k == which.max(acc))) +  
 labs(x = 'K', y = 'Accuracy', title = 'KNN Accuracy for different values of K') +  
 scale\_x\_continuous(breaks = 1:20) +  
 coord\_cartesian(ylim = c(min(acc), max(acc))) +  
 guides(fill = FALSE)

 ## 4.12

This problem involves writing functions. (a) Write a function, Power(), that prints out the result of raising 2 to the 3rd power. In other words, your function should compute 2^3 and print out the results.

Hint: Recall that x^a raises x to the power a. Use the print() function to output the result.

Power <- function() {  
 print(2^3)  
}  
Power()

## [1] 8

1. Create a new function, Power2(), that allows you to pass any two numbers, x and a, and prints out the value of x^a. You can do this by beginning your function with the line

Power2 <- function(x, a) {  
 print(x^a)  
}  
Power2(3, 8)

## [1] 6561

1. Using the Power2() function that you just wrote, compute

Power2(10,3)

## [1] 1000

Power2(8,17)

## [1] 2.2518e+15

Power2(131,3)

## [1] 2248091

1. Now create a new function, Power3(), that actually returns the result x^a as an R object, rather than simply printing it to the screen. That is, if you store the value x^a in an object called result within your function, then you can simply return() this result, using the following line: The line above should be the last line in your function, before the } symbol.

Power3 <- function(x , a) {  
 result <- x^a  
 return(result)  
}

1. Now using the Power3() function, create a plot of f(x) = x .

x <- 1:10  
plot(x, Power3(x, 2),log = "xy", xlab = "Log of x", ylab = "Log of x^2", main = "Log of x^2 vs Log of x")



1. Create a function, PlotPower(), that allows you to create a plot of x against x^a for a fixed a and for a range of values of x. For instance, if you call

PlotPower <- function(x, a) {  
 plot(x, Power3(x, a), xlab = "Log of x", ylab = paste("Log of x",'^',a))  
}  
  
PlotPower(1:10, 3)

