Logistic Regression Approach to Predicting Prostate Cancer from Baseline Exam Measurements

—

San Diego State University
STAT 794: Statistical Communication in Data Science

November 12, 2020

Abstract

This analysis aims to determine if baseline exam measurements can predict whether a tumor has penetrated the prostatic capsule. A logistic regression model was built to predict capsular penetration using the results of the digital rectal exam, prostatic specific antigen (PSA) level, and total Gleason score. With this model an individual with left uniolbar nodule is 2.2 times, an individual with right uniolabr nodule is 4.7 times, and an individual with bilobar nodule is 4.2 times more likely to have capsular penetration. For every unit increase in PSA level an individual will be 3% more likely to have capsular penetration. For every unit increase in total Gleason score, a patient will be 2.7 times more likely to have capsular penetration.

1 Introduction

Prostate cancer is a type of cancer that develops on the prostate gland, a small walnut shaped gland in the pelvis of men. It is second most common cancer in U.S. men (after skin cancer) with about 1 in 9 men being diagnosed in their lifetime. Due to the pervasiveness of this disease, early detection is very important to being able to actively monitor and treat the disease. Screenings are routinely done to test for the disease even if their are no symptoms. One of these screenings is a protein-specific antigen (PSA) blood test which tests for the level of PSA in the blood. A low PSA is a sign of prostate health, but a rise in PSA may be a sign of something wrong, specifically with the prostate. Another type of screening that is routinely done is the digital rectal examine (DRE). A doctor performs this test to feel for an abnormal shape or thickness to the prostate. Additionally, a physician may do further screening by collecting prostate cells through a biopsy. The Gleason score is a scale measuring the abnormality of cells from core biopsies. Larger values suggest higher risk of cancer.

These screenings are all aimed at catching prostate cancer as early as possible and to streamline treatment. Once the tumor has penetrated the prostatic capsule, more aggressive treatment options, such as surgery, may be necessary to treat the patient. We will build a logistic regression model using the baseline exam measurements in order to predict whether a patient will have penetration of the prostatic capsule. This will allow physicians to understand what type of patient is more likely to have prostatic capsule penetration.

2 Methods

The data was collected from the Ohio State University Comprehensive Cancer Center. Variables include one patient identifier, our response variable, capsule, and five baseline exam measurements or predictor variables. The baseline exam measurements, predictor variables include, age of subject, race of subject, results of the digital rectal exam, detection of capsular involvement, prostatic specific antigen (PSA) value, and total Gleason score. 153 of 380 subjects had a cancer that penetrated the capsule. Table 2 shows the data and variables in more detail. We will utilize this data to build a logistic regression model. This regression model will be used to predict whether a tumor has penetrated the capsule based on the baseline exam measurements. All analysis was performed in R Studio with R 3.6.2.

3 Results

3.1 Exploratory Data Analysis

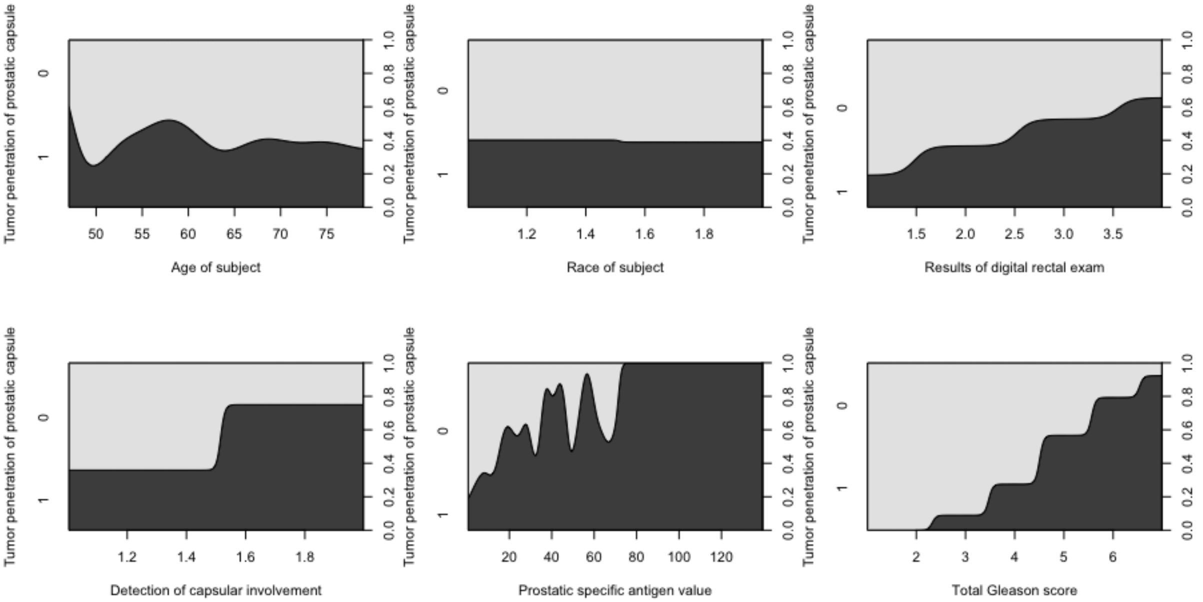

Figure 1: *Conditional density plots of all predictor variables to be considered in the regression. Race of subject does not appear to have a relationship with tumor penetration of prostatic capsule. Results of digital rectal exam, detection of capsular involvement, prostatic specific antigen value, and total Gleason score appear to have a relationship with tumor penetration of prostatic capsule. Age may have a relationship with tumor penetration of prostatic capsule.*

This analysis aims to predict whether a tumor has penetrated the prostatic capsule from several baseline exam measurements. Because tumor penetration is a binary response variable, we will use logistic regression. First we should note that 153 of the 380 subjects included in this data set had prostate cancer that penetrated the capsule. Because there is not only a small number of subjects with tumor penetration of the capsule, we can use logistic regression to build our model.

Now we can investigate the predictor baseline exam measurement variables. To begin our analysis, in Figure 1, we created conditional density plots to understand the relationship of the predictor variables and whether there is penetration of the prostatic capsule. Race of subject is categorized as white or black. There does not appear to be a relationship between the race of the subject and whether they will have penetration of the prostatic capsule as seen by the constant density. It is difficult to determine whether or not the age of the subject has a relationship with tumor penetration, so this variable will need to be investigated further. The results of digital rectal exam, detection of capsular involvement, prostatic specific antigen value, and total Gleason score appear to have a relationship with tumor penetration of prostatic capsule. We anticipate these variables contributing to our model.

To investigate the predictor variables further, we calculate the Cohen's d value to measure the effect size. Race has a Cohen's d value of $d = 0.015$, leading us to conclude that the effect size is trivial and there is not a relationship between a subject's age and whether a subject will have tumor penetration of the prostatic capsule. Similarly, the small Cohen's d value of $d = 0.08$ leads us to conclude that there is not a meaningful relationship between a subject's race and tumor penetration.

3.2 Model Fitting and Inferences

	Dependent variable:
	Tumor penetration of prostatic capsule
dpros2	0.77** (0.07, 1.47)
dpros3	1.55*** (0.83, 2.28)
dpros4	1.43*** (0.55, 2.31)
psa	0.03*** (0.01, 0.05)
gleason	1.00*** (0.68, 1.31)
Constant	-8.14*** (-10.22, -6.07)
Observations	377
Akaike Inf. Crit.	393.22
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 1: Regression inferences from AIC selection without age with 95% confidence intervals. Variables include results of the digital rectal exam, prostatic specific antigen value, and total Gleason score.

Having explored the variables, we begin building our model fitting by looking at a model which includes all variables and their interactions. Using AIC selection on the full model, the new model includes the age of the subject, results of the digital rectal exam, prostatic specific antigen value, total Gleason score, the interaction of age and the results of the digital rectal exam, and the interaction age and total Gleason score. Figure Table 3 contains the regression inference and 95%

confidence intervals for the predictor variables in the AIC selected model. Though this model has the lowest AIC score, the inclusion of age as a variable leads us to consider whether this is truly the best model. Remember that the Cohen's d value for age was $d = 0.08$, leading us to conclude that the effect size was negligible. For this reasons we will investigate models that do not include age to see if it would be a better fit.

To investigate models further we will use the AIC selected model and see how the AIC is effected when the age of subjects is removed. Table 1 shows the regression inference for this model. When we removed age, and are left with the results of digital rectal exam, prostatic antigen value, and the total Gleason score our model has an AIC of 393.22, which is not much larger than the AIC selected model with an AIC score of 391.07. All variables included in the model are statistically significant at the $\alpha = 0.05$ level. Additionally, because this model does not include interactions, the model will be easier to interpret. For these reasons, we will proceed with this model and check the diagnostics to confirm that it meets our assumptions and will be usable for our predictions.

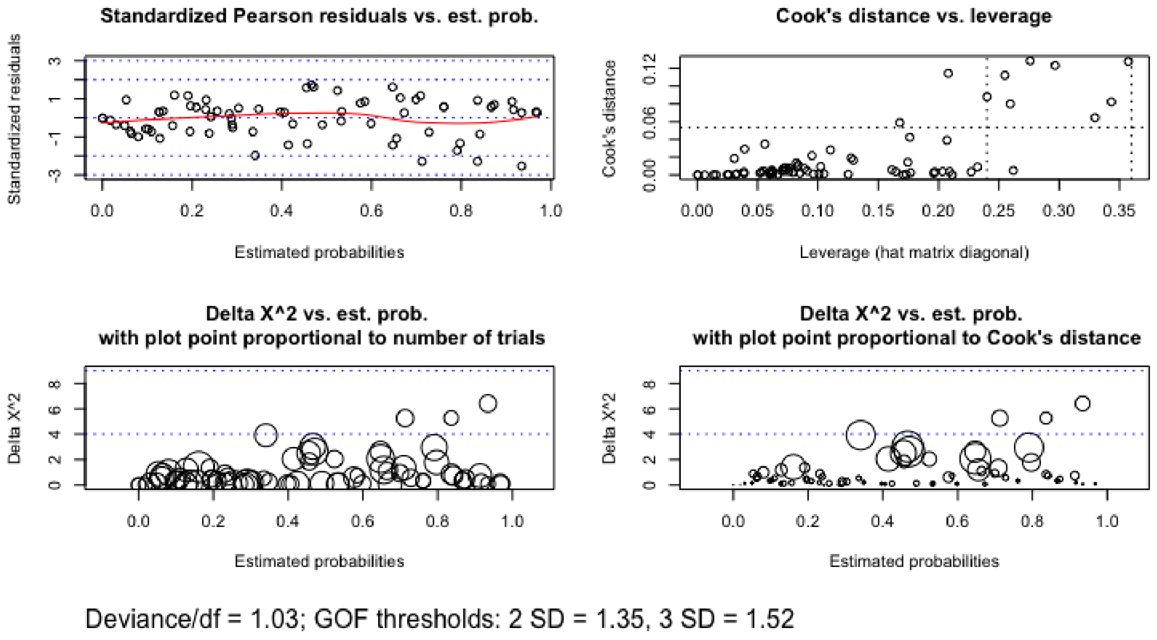

Figure 2: *Plots used to asses our model's performance. We conclude that our model meets the assumptions and can be used to model our data.*

To evaluate our model, we will look at several things. First, looking at Figure 2 we have a plot of standardized residuals. Looking at the red line, we see that it is relatively linear and the data does not seem to have a specific pattern. There are a few exploratory variable patterns of note, but after looking at them, we decided to include them as there was no sufficient reason to remove them. Our data set is rather small so we do not want to remove variables unless essential. A Hosmer-Lemeshow test (HL test) for goodness of fit was also preformed and we conclude that our model is a good fit for the data.

In addition, we also calculated our model's sensitivity and specificity by testing a subset of our data. Using our final model on the test subset, we have a sensitivity, percentage of true patient's with capsular penetration or true positives, of 62%. This is not particularly accurate, however our specificity, or true negatives, is much better at 85%. To visualize the specificity and sensitivity we created a ROC curve as seen in Figure 3. We see that our model is a pretty good classifier. The area under the cure (AUC) is 0.8 which again makes us conclude that our model is a pretty good classifier.

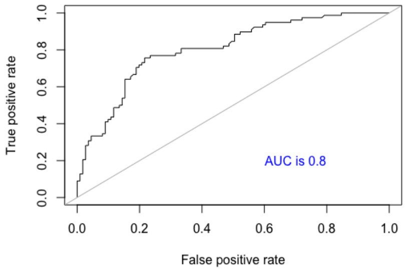

Figure 3: *ROC plot of our final model (which includes total Gleason score, PSA, and results of digital rectal exam) with a calculated AUC=0.8. We conclude that our model is a good classifier for whether a patient will have capsule penetration.*

We can now use our final model to make interpretations., specifically we will discuss the odds ratios associated with the variables included in the model. The variables included in our final model include the results of the digital rectal exam, prostatic antigen value, and the total Gleason score. Using the regression inferences in Table 1 and the odds ratios we can make inferences about a patient's odds of having tumor penetration of prostatic capsule. To interpret the digital rectal exam inferences, we must note that the baseline is a patient with no nodules. An individual with unilobar nodule (left) is 2.2 times more likely to have capsular penetration. We are 95% confident that an individual with left uniolabr nodule will be between 1.1 and 4.3 time more likely to have capsular penetration. Additionally, an individual with right unilobar nodule is 4.7 (95% CI: (2.3, 9.8)) times more likely to have capsular penetration. And finally, an individual with bilobar nodule is 4.2 (95% CI: (1.7, 10.1)) times more like to have capsular penetration compared to patients with no nodules.

Now looking at PSA, for every unit increase in prostatic specific antigen and individual will be 3% more likely to have capsular penetration. We are 95% confident that a unit increase in PSA will result in a 1% to 5% increase in the likelihood of penetration of prostatic capsule. Finally, for every unit increase in total Gleason score, a patient will be 2.7 times more likely to have capsular involvement. We are 95% confident that a unit increase in total Gleason score will result in 2.0 to 3.7 times more likely to have capsular penetration.

4 Conclusions

The goal of this analysis is to determine if baseline exam measurements from routine medical screenings can predict whether a tumor has penetrated the prostatic capsule. A logistic regression model was built including the results of the digital rectal exam, prostatic specific antigen level, and total Gleason score to predict whether there was penetration of the capsule. An interaction between PSA and the total Gleason score was also included in the model. No transformations were applied to allow for ease of interpretation.

Using our model, we were able to conclude the following:

- An individual with unilobar nodule (left) is 2.2 times more likely to have capsular penetration, an individual with unilobar nodule (right) is 4.7 times more likely to have capsular penetration, and an individual with bilobar nodule is 4.2 times more like to have capsular penetration compared to patients with no nodules.
- For every unit increase in prostatic specific antigen and individual will be 3% more likely to have capsular penetration.
- For every unit increase in total Gleason score, a patient will be 2.7 times more likely to have capsular involvement.

Using the proposed model, physicians will be able to plan treatments more effectively and provide better treatment to their patients. Being able to better assess the odds of someone has capsular penetration of the prostate will help physicians be more aggressive or restraint, depending on the patient's baseline exam values.

There are several limitations to our study to be noted. First, our sample size is only 380 subjects. There were several exploratory variable patterns that may have been influential that we choose to include for this reasons. Future research studies should investigate whether this model holds for larger sample sizes or exploratory variable patterns become more troublesome as the data set grows.

References

[1] Kutner, Michael H., et al. (2018.) *Applied Linear Regression Models*. McGraw-Hill Education, Asia.

[2] https://www.urologyhealth.org/urology-a-z/p/prostate-cancer

Appendix: Supplementary Figures

Variable	Description	Coding
ID	Identification code	integer
capsule	Tumor penetration of prostatic capsule	1=penetration
age	Age of subject	years
race	Race of the subject	1=white, 2=black
dpros	Results of the digital rectal exam	1=no nodule 2=uniolab nodule (left) 3=unilobar nodule (right) 4=bilobar nodule
dcaps	Detection of capsular involvement	1=no, 2=yes
psa	Prostatic Specfic Antigen value	*mg/ml*
gleason	Total Gleason score	scale of 1-10

Table 2: *Summary of variables included in data set and the coding in the data set.*

	Dependent variable:
	Tumor penetration of prostatic capsule
age	0.27(-0.07,0.62)
dpros2	8.97** (1.22, 16.72)
dpros3	1.24(-6.83,9.30)
dpros4	3.72(-6.72,14.16)
psa	0.03*** (0.01, 0.05)
gleason	3.49** (0.07, 6.91)
age:dpros2	-0.12** $(-0.24, -0.01)$
age:dpros3	0.01(-0.11,0.13)
age:dpros4	-0.03(-0.19,0.12)
age:gleason	-0.04(-0.09,0.01)
Constant	-26.83** $(-50.49, -3.17)$
Observations	377
Akaike Inf. Crit.	391.07
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 3: Regression inferences from AIC selection with 95% confidence intervals. Variables include age of the subject, results of the digital rectal exam, prostatic specific antigen value, total Gleason score, the interaction of age and the results of the digital rectal exam, and the interaction of prostatic specific antigen and total Gleason score.

Appendix: R-Code

```
library(corrplot)
library(MASS) #for stepAIC
library(effsize) # for Coehn's d
library(stargazer)
library(binom)
library(ISLR)
library(ROCR)


## Load in prostate data set and take initial looks
Prostate <- read.delim("~/Downloads/prostate.txt")
dim(Prostate)
#[1] 380 obs of 8 variables
sum(is.na(Prostate)) # missing data: 3 in race
Prostate<-na.omit(Prostate); dim(Prostate) # after removing missing data; 377 obs of 8 vars
Prostate$dcaps = as.factor(Prostate$dcaps)
Prostate$dpros = as.factor(Prostate$dpros)
attach(Prostate)


#############
#    EDA    #
#############
#Conditional density plots graphically present how a binary response changes over a covariate.
par(mfrow=c(2,3))
cdplot(factor(capsule)~age,
       ylab="Tumor penetration of prostatic capsule", xlab="Age of subject")
cdplot(factor(capsule)~factor(race),
       ylab="Tumor penetration of prostatic capsule", xlab="Race of subject")
cdplot(factor(capsule)~factor(dpros),
       ylab="Tumor penetration of prostatic capsule", xlab="Results of digital rectal exam")
cdplot(factor(capsule)~factor(dcaps),
       ylab="Tumor penetration of prostatic capsule", xlab="Detection of capsular involvement")
cdplot(factor(capsule)~psa,
       ylab="Tumor penetration of prostatic capsule", xlab="Prostatic specific antigen value")
cdplot(factor(capsule)~factor(gleason),
       ylab="Tumor penetration of prostatic capsule", xlab="Total Gleason score")
dev.off()

table(capsule) #151 show penetration, 226 show no penetration
signif(prop.table(table(capsule)), digits = 3) #not too much 0 or 1 which allows us to fit a lo
table(capsule, race) #204 white individuals have no penetration, 22 black individuals have no p
table(capsule, dpros)
table(capsule, dcaps)
table(capsule, gleason)

plot(factor(capsule), psa, col="cyan", varwidth=T,
     ylab="Prostatic Specific Antigen value (mg/ml)" ,xlab="Tumor penetration of prostatic caps
plot(factor(capsule), age, col="cyan", varwidth=T,
     ylab="Age of Subject" ,xlab="Tumor penetration of prostatic capsule")

# Cohen’s d (cohen.d function) to evaluate the relationship between capsule and variables.
cohen.d(race, factor(capsule))  # d = 0.015; trival (not significnat)
cohen.d(psa, factor(capsule))  # d = -0.697; meadium to large effect (significant)
cohen.d(age, factor(capsule)) # d=0.08  (negligible)
cohen.d(gleason, factor(capsule)) #d=-1.026 (large)





## Model building
# Stepwise model selection: include interactions, consider stepAIC for first pass
fit.full=glm(capsule~.*., family=binomial(link=logit), data=Prostate)
summary(fit.full)
stepAIC(fit.full)
#model includes psa, gleason, race, and dpros
#      identify a simpler model by being strict with interaction terms.
#      Is it that much worse than best stepwise model?
fit.1=glm(formula = capsule ~ age + dpros + psa + gleason + age:dpros +
     age:gleason, family = binomial(link = logit), data = Prostate)#from stepAIC(fit.full)
summary(fit.1)
stargazer(fit.1, title="Regression inferences from AIC selection.",
         label="reginf",
         align=TRUE,  ci=TRUE, ci.level=0.95, single.row=TRUE, omit.stat=c("LL", "ser", "f"),
         digits=2)

fit.1.5=glm(formula = capsule ~ dpros + psa + gleason, family = binomial(link = logit), data =
summary(fit.1.5)
stargazer(fit.1.5, title="Regression inferences from AIC selection with age removed.",
          label="reginf",
          align=TRUE,  ci=TRUE, ci.level=0.95, single.row=TRUE, omit.stat=c("LL", "ser", "f"),
          digits=2)

fit.2=glm(capsule~gleason+psa+dpros+dcaps+psa:gleason+dpros:dcaps, family=binomial(link=logit)
summary(fit.2) #not great AIC: 399.44

fit.3=glm(capsule~gleason+psa+dpros+psa:gleason, family=binomial(link=logit), data=Prostate)
```

```
summary(fit.3) # pretty good AIC: 395.13

fit.4=glm(capsule~gleason+dpros, family=binomial(link=logit), data=Prostate)
summary(fit.4) #bad AIC: 401.35

fit.5=glm(capsule~gleason+psa:gleason+dpros+dcaps, family=binomial(link=logit), data=Prostate)
summary(fit.5) #AIC 394.09

fit.6=glm(capsule~gleason+psa:gleason+dpros, family=binomial(link=logit), data=Prostate)
summary(fit.6) #AIC: 393.5

fit.7=glm(capsule~gleason+psa+psa:gleason+dpros+dcaps, family=binomial(link=logit), data=Prosta
summary(fit.7) #AIC 395.7


## Model evaluation
# The functions for residual analysis we proposed using are as follows:
one.fourth.root=function(x){
  x^0.25
}
# make sure examine.logistic.reg.R is in your working directory or you have the right path spe
#source("examine.logistic.reg.R")

# Consider a model of PSA, Gleason score, and Results of digital rectal exam
dat.glm <- glm(capsule ~ psa+gleason+dpros, family = binomial, data = Prostate)
dat.mf <- model.frame(dat.glm)

## Covariate pattern: too many EVPs!
w <- aggregate(formula = capsule ~ psa+gleason+dpros, data = Prostate, FUN = sum)
n <- aggregate(formula = capsule ~ psa+gleason+dpros, data = Prostate, FUN = length)
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
dim(w.n)
#[1] 342 6

# Create EVPs by binning continuous covariates
g = 5 # number of categories
psa_interval = cut(psa, quantile(psa, 0:g/g), include.lowest = TRUE)  # Creates factor with le
levels(psa_interval)

w <- aggregate(formula = capsule ~ psa_interval+gleason+dpros, data = Prostate, FUN = sum)
n <- aggregate(formula = capsule ~ psa_interval+gleason+dpros, data = Prostate, FUN = length)
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
mod.prelim1 <- glm(formula = capsule/trials ~ psa_interval+gleason+dpros,
                   family = binomial(link = logit), data = w.n, weights = trials)
save1=examine.logistic.reg(mod.prelim1, identify.points=T, scale.n=one.fourth.root, scale.cook
w.n.diag1=data.frame(w.n, pi.hat=round(save1$pi.hat, 2), std.res=round(save1$stand.resid, 2),
                     cookd=round(save1$cookd, 2), h=round(save1$h, 2))
p=length(mod.prelim1$coef) # number of parameters in model (# coefficients)
ck.out=abs(w.n.diag1$std.res)>2 | w.n.diag1$cookd>4/nrow(w.n) | w.n.diag1$h > 3*p/nrow(w.n)
extract.EVPs=w.n.diag1[ck.out, ]
extract.EVPs


#source("HLtest.R")
HL = HLTest(fit.1.5, 4)
# HL test output: Y0 are successes, Y1 are failures
cbind(HL$observed, round(HL$expect, digits=1))

#ROC curve
# Split data into training and testing sets and illustrate sensitivity and specificity
c = 0.5 # probability cutoff for predicting a default
set.seed(1)  # set the random number generator seed
p = 0.5 # percentage split for training and testing sets
n = dim(Prostate)[1]  # sample size
train = sample(n, p*n)  # random sample percentage out of n; this creates the index list
length(train); head(train) # take a look at the 'train' variable index

# Fit logistic regression model on the training subset
fit_train = glm(capsule~gleason+dpros+psa, family=binomial(link=logit), data=Prostate, subset =
summary(fit_train)

test_probs1.5 = predict.glm(fit.1.5, Prostate, type="response")[-train]
test_class = (test_probs1.5 > c)

table(test_class, Prostate$capsule[-train], dnn=c("Predicted", "Truth"))  # cross-classificatio
sum(as.numeric(test_class) == (as.numeric(Default$capsule[-train])-1))/(n-n*p) # accuracy
# Rows are predicted default status, columns are true default status
#test_class   No  Yes
#FALSE 94  30
#TRUE    17   48
#sensitivity = 48/(30+48)  # percentage of true defaulters correctly identified: 62%  (true pos
#specificity = 94/(94+17) # percentage of non-defaulters correctly identified: 85% (true negati
# Function that will compute sensitivity and specificity at any given cutoff
se.sp <- function (cutoff, pred){
  sens <- performance(pred,"sens")
  spec <- performance(pred,"spec")
  num.cutoff <- which.min(abs(sens@x.values[[1]] - cutoff))
  return(list(Cutoff=sens@x.values[[1]][num.cutoff],
              Sensitivity=sens@y.values[[1]][num.cutoff],
              Specificity=spec@y.values[[1]][num.cutoff]))
```

```
}
pred = prediction(test_probs1.5, Prostate[-train,"capsule"])  # Prediction class from ROCR pack
se.sp(0.5, pred) # Sensitivity and specificity at 0.5 cutoff
# What if increase cut-off?
se.sp(0.8, pred)
# What if decrease cut-off?
se.sp(0.3, pred)


# ROC curve illustration, presenting cutpoints on a colorized curve
pred = prediction(test_probs1.5, Prostate[-train,"capsule"])
ROCRperf = performance(pred, "tpr", "fpr")
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7),
     xlab="False positive rate (1-specificity)", ylab="True positive rate (sensitivity)")
abline(a=0, b= 1, col="gray")
arrows(0.3,0.2,0.2,0.2,length=0.1)
text(0.3,0.2,"coin flip; no better than random", cex=1, pos=4)
# ROC curve for reporting
plot(ROCRperf)
abline(a=0, b= 1, col="gray")
# calculating AUC
auc1 = performance(pred,"auc")
# convert S4 class to vector
auc1 = unlist(slot(auc1, "y.values"))
text(0.7,0.2,paste("AUC is", signif(auc1, digits=2)), col="blue")



###Evaluation of fit.3=glm(capsule~gleason+psa+dpros+psa:gleason, family=binomial(link=logit),

# The functions for residual analysis we proposed using are as follows:
one.fourth.root=function(x){
  x^0.25
}

dat.mf.3 <- model.frame(fit.3)

## Covariate pattern: too many EVPs!
w <- aggregate(formula = capsule ~ gleason+psa+dpros+psa:gleason, data = Prostate, FUN = sum)
n <- aggregate(formula = capsule ~ gleason+psa+dpros+psa:gleason, data = Prostate, FUN = length
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
dim(w.n)
#[1] 342 6

# Create EVPs by binning continuous covariates
```

```
g = 5 # number of categories
psa_interval = cut(psa, quantile(psa, 0:g/g), include.lowest = TRUE)  # Creates factor with lev
levels(psa_interval)

w <- aggregate(formula = capsule ~ psa_interval+gleason+dpros+psa_interval:gleason, data = Pros
n <- aggregate(formula = capsule ~ psa_interval+gleason+dpros+psa:gleason, data = Prostate, FUN
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
mod.prelim1 <- glm(formula = capsule/trials ~ psa_interval+gleason+dpros+psa_interval:gleason,
                   family = binomial(link = logit), data = w.n, weights = trials)
save1=examine.logistic.reg(mod.prelim1, identify.points=T, scale.n=one.fourth.root, scale.cook
w.n.diag1=data.frame(w.n, pi.hat=round(save1$pi.hat, 2), std.res=round(save1$stand.resid, 2),
                     cookd=round(save1$cookd, 2), h=round(save1$h, 2))
p=length(mod.prelim1$coef) # number of parameters in model (# coefficients)
ck.out=abs(w.n.diag1$std.res)>2 | w.n.diag1$cookd>4/nrow(w.n) | w.n.diag1$h > 3*p/nrow(w.n)
extract.EVPs=w.n.diag1[ck.out, ]
extract.EVPs


#source("HLtest.R")
HL = HLTest(fit.3, 4)
# HL test output: Y0 are successes, Y1 are failures
cbind(HL$observed, round(HL$expect, digits=1))


###Evaluation of fit.7=glm(capsule~gleason+psa+psa:gleason+dpros+dcaps, family=binomial(link=lo


# The functions for residual analysis we proposed using are as follows:
one.fourth.root=function(x){
  x^0.25
}

dat.mf.7 <- model.frame(fit.7)

## Covariate pattern: too many EVPs!
w <- aggregate(formula = capsule ~ gleason+psa+psa:gleason+dpros+dcaps, data = Prostate, FUN =
n <- aggregate(formula = capsule ~ gleason+psa+psa:gleason+dpros+dcaps, data = Prostate, FUN =
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
dim(w.n)
#[1] 342 7

# Create EVPs by binning continuous covariates
g = 5 # number of categories
psa_interval = cut(psa, quantile(psa, 0:g/g), include.lowest = TRUE)  # Creates factor with lev
levels(psa_interval)
```

```
w <- aggregate(formula = capsule ~ psa_interval+gleason+dpros+psa_interval:gleason+dcaps, data
n <- aggregate(formula = capsule ~ psa_interval+gleason+dpros+psa:gleason+dcaps, data = Prosta
w.n <- data.frame(w, trials = n$capsule, prop = round(w$capsule/n$capsule,2))
mod.prelim1 <- glm(formula = capsule/trials ~ psa_interval+gleason+dpros+psa_interval:gleason+
                   family = binomial(link = logit), data = w.n, weights = trials)
save1=examine.logistic.reg(mod.prelim1, identify.points=T, scale.n=one.fourth.root, scale.cook
w.n.diag1=data.frame(w.n, pi.hat=round(save1$pi.hat, 2), std.res=round(save1$stand.resid, 2),
                     cookd=round(save1$cookd, 2), h=round(save1$h, 2))
p=length(mod.prelim1$coef) # number of parameters in model (# coefficients)
ck.out=abs(w.n.diag1$std.res)>2 | w.n.diag1$cookd>4/nrow(w.n) | w.n.diag1$h > 3*p/nrow(w.n)
extract.EVPs=w.n.diag1[ck.out, ]
extract.EVPs


#source("HLtest.R")
HL = HLTest(fit.1.5, 4)
# HL test output: Y0 are successes, Y1 are failures
cbind(HL$observed, round(HL$expect, digits=1))
```