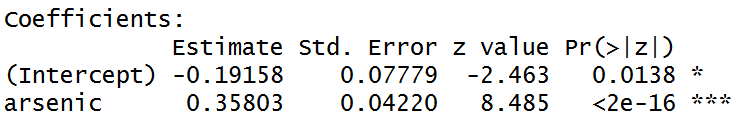
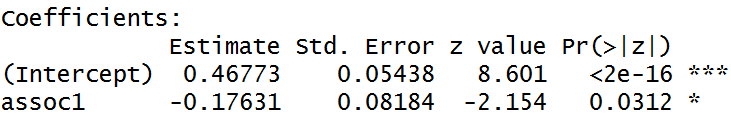
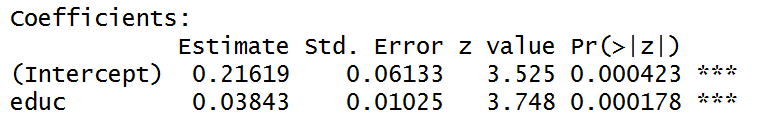
Homework 4

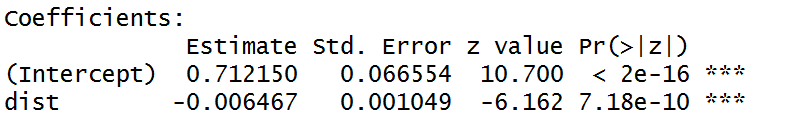
Yue Peng

1. Construct a good logistic regression model predicting the decision to switch wells as a function of the 4 predictors (arsenic, distance, association and education) on the training data. Consider potential transformations of continuous variables and possible interactions.

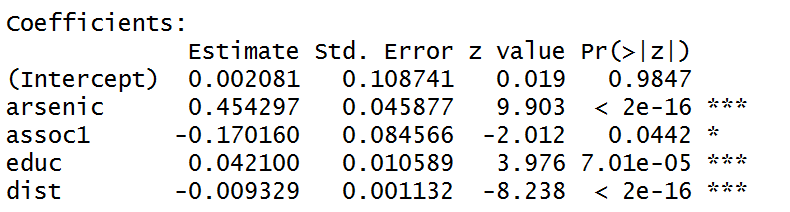




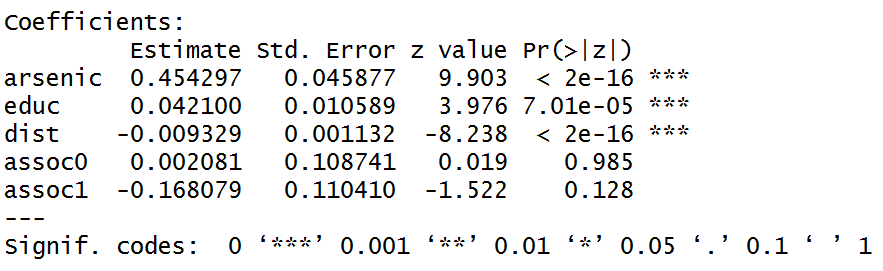




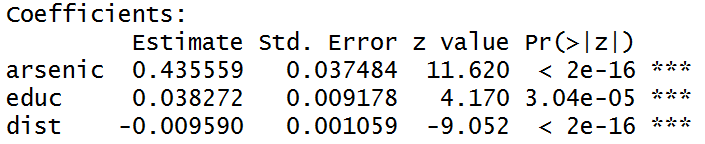
After fitting each predictor into the logistic regression model, we decided to use them all.

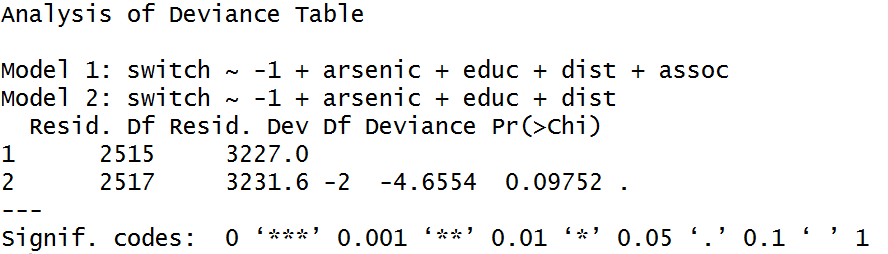


We would try to remove the “intercept” term since the p-value was way larger than 0.05.



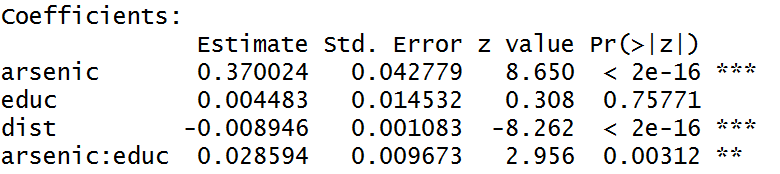
We would try to remove the “assoc” term since the p-value was larger than 0.05.

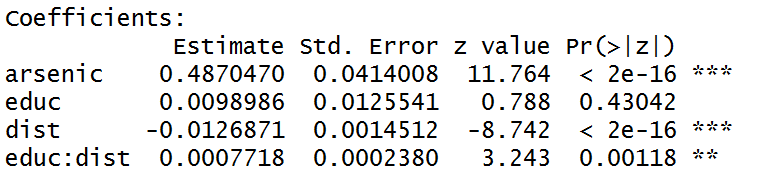


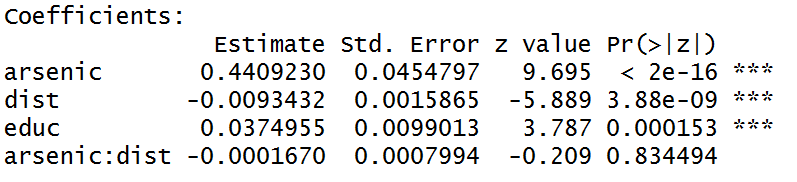


With the ANOVA table, we decided to use “arsenic”, “educ” and “dist” as predictors since the p-value was larger than 0.05.

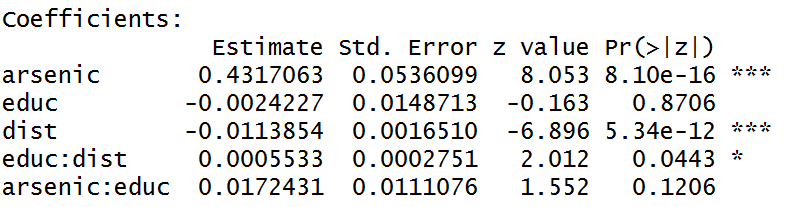
The next step was to test the interaction term.

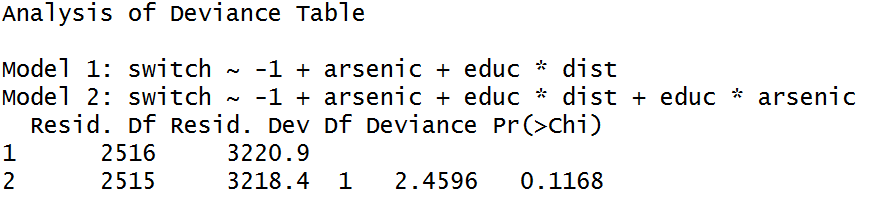




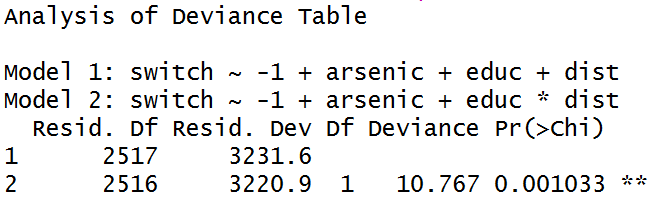


We decided to test the interaction effect of “educ” with “arsenic” and with “dist”.





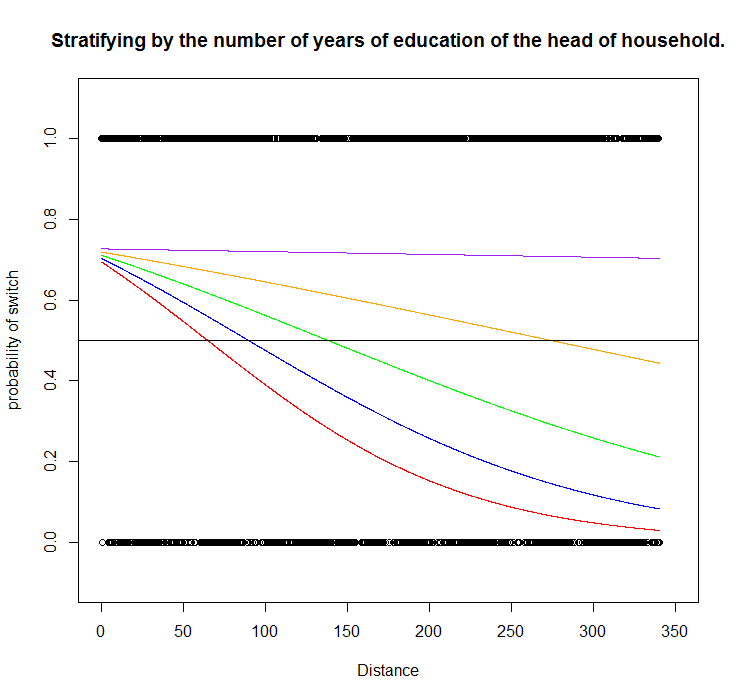
The model with only “educ\*dist” was better since the p-value was larger than 0.05.



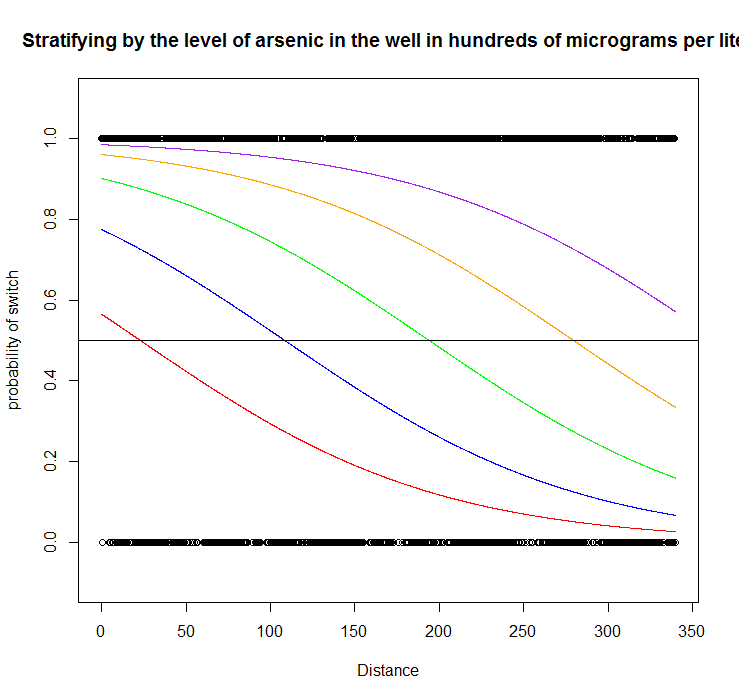
The model including the “educ\*dist” was better than model without interaction term since the p-value was less than 0.05.

Thus, our final model was

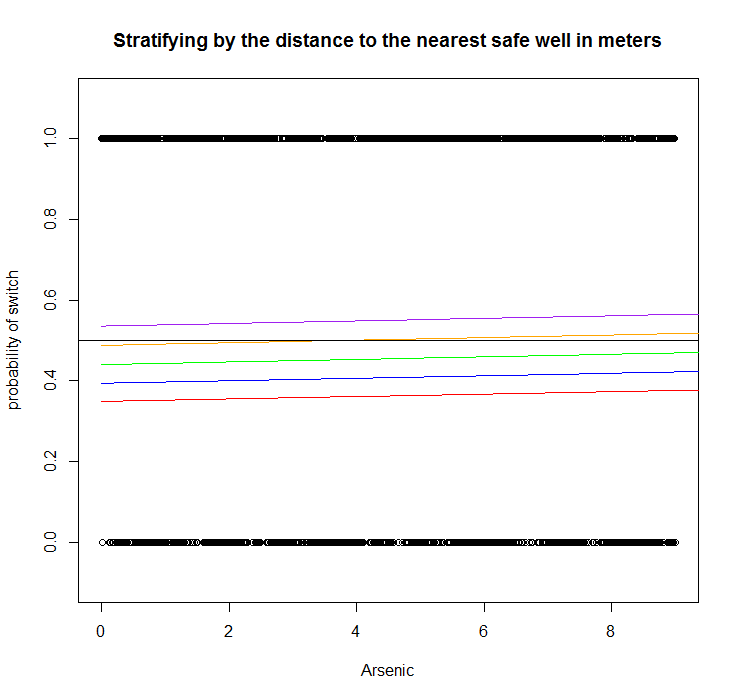
1. Compute and graph the predicted probabilities stratifying by the predictors. You could do this using graph such as in the papers we discussed in class or by using contour plots which would allow you to graph two continuous predictors on the same plot. You can array different lines and plots to try to put this all on one sheet or you can spread across different plots. See what works best.



For the curves above, the “educ” kept increases while fixing the “arsenic” as the mean value of the training dataset. The curve with longer “educ” went above the one with shorter “educ”. It meant that the larger number of years of education of the head of household, the more likely for the household to switch wells.



For the curves above, the “arsenic” kept increases while fixing the “educ” as the mean value of the training dataset. The curve with higher “arsenic” went above the one with lower “arsenic”. It meant that the higher level of arsenic in the well in hundreds of micrograms per liter, the more likely for the household to switch wells.



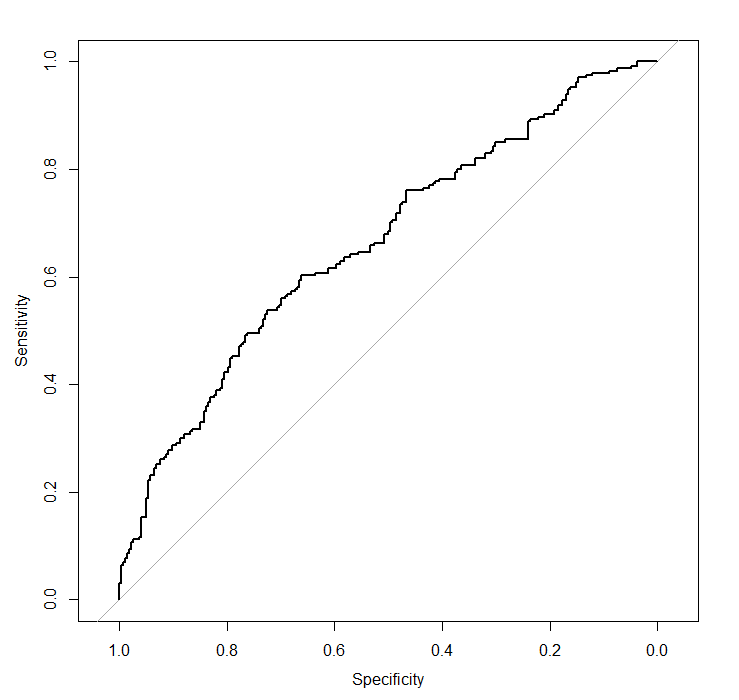
For the curves above, the “educ” kept increases while fixing the “dist” as the mean value of the training dataset. The curve with longer “educ” went above the one with shorter “educ”. It meant that the larger number of years of education of the head of household, the more likely for the household to switch wells.

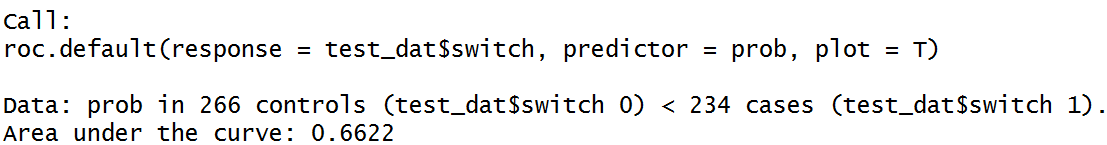
1. Compute the confusion matrix on the test data using p = 0.5 as a cutoff and discuss what this tells you about the predictive model you have constructed (e.g. sensitivity, specificity, error rate, etc.)

|  |  |  |
| --- | --- | --- |
|  | True switch | True no switch |
| Predicted switch | 210 (TP) | 208 (FP) |
| Predicted no switch | 24 (FN) | 58 (TN) |

According to the high sensitivity and low specificity, this model performed well in predicting the household has switched wells but badly in predicting the household has not switched wells.

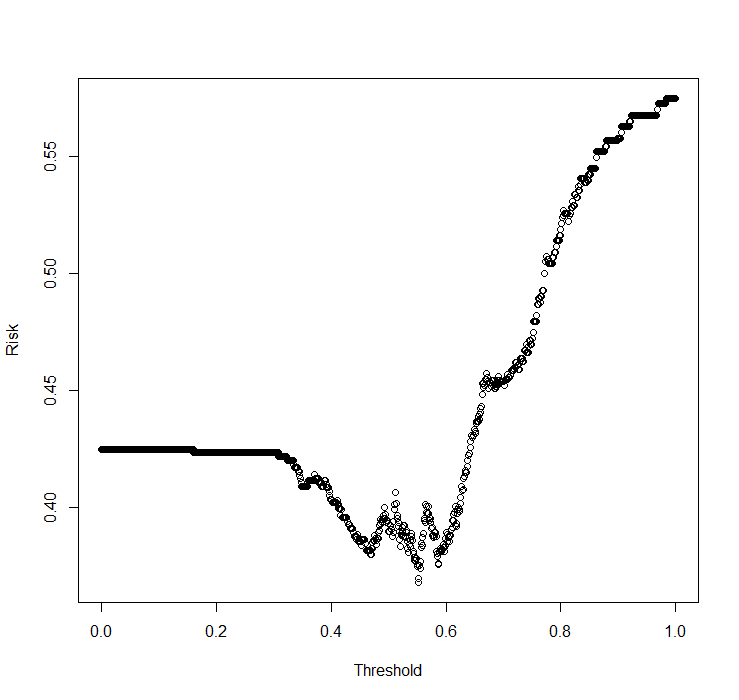
1. Construct an ROC plot and compute the area under the ROC curve.





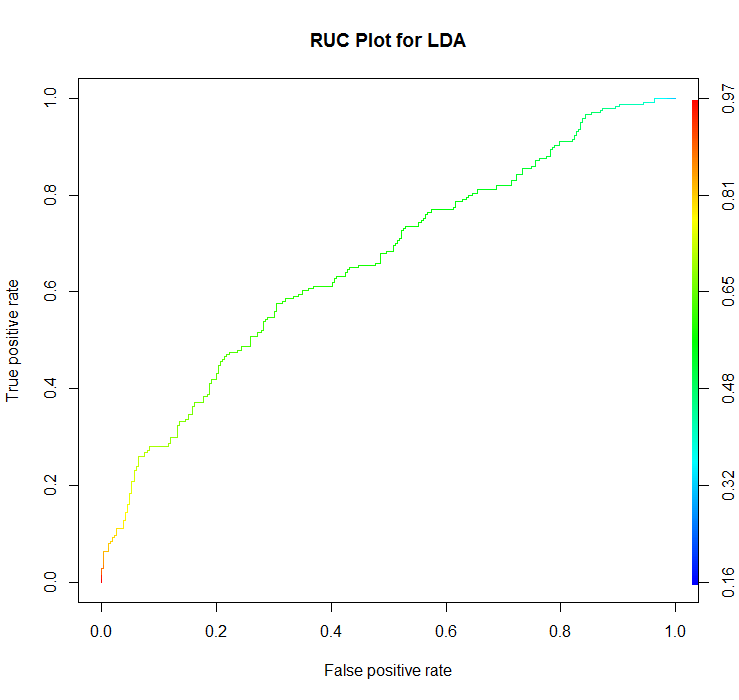
The ROC curve was shown above, and the AUC was 0.6622, which did relatively well in prediction.

1. What does this curve tell you about choice of threshold that balances sensitivity with specificity (i.e., how would you balance risk of switching and not switching?)



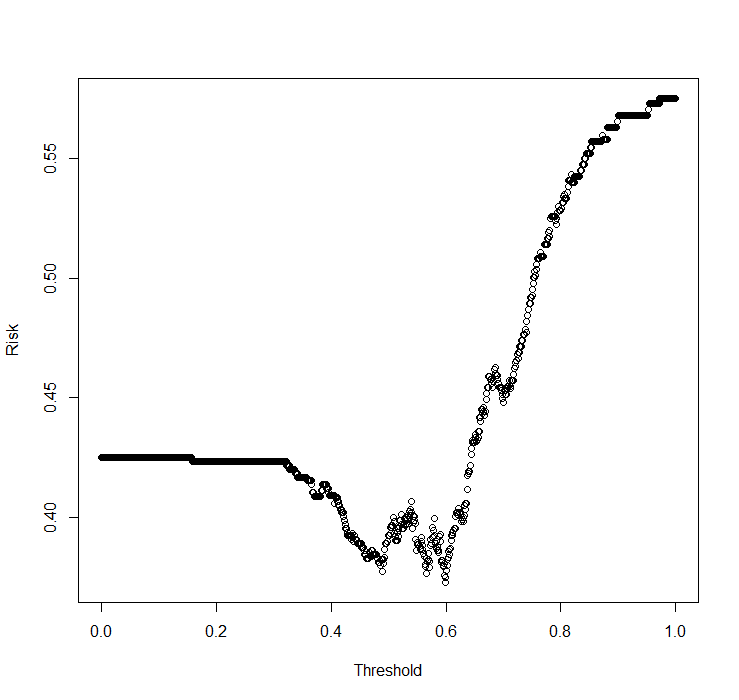
The threshold with the least risk (0.368) was 0.55. If refitting this into our previous model, our sensitivity would be changed into 0.76 and specificity was 0.45. Compared to the previous model, the predictive ability increased since the specificity has increased a lot while sacrifice a little sensitivity.

1. Repeat this analysis using linear discriminant analysis, quadratic discriminant analysis and K nearest neighbor with K = 1 and K = 5. For discriminant analysis, note that the predict function returns 3 elements:*class* is a binary indicator as to whether the posterior probability is greater than 0.5, *posterior* gives the posterior predictive probability and *x* contains the linear discriminant for the LDA function (missing for QDA). Note that for KNN, you will only get classifications, not probabilities.
2. Linear Discriminant Analysis



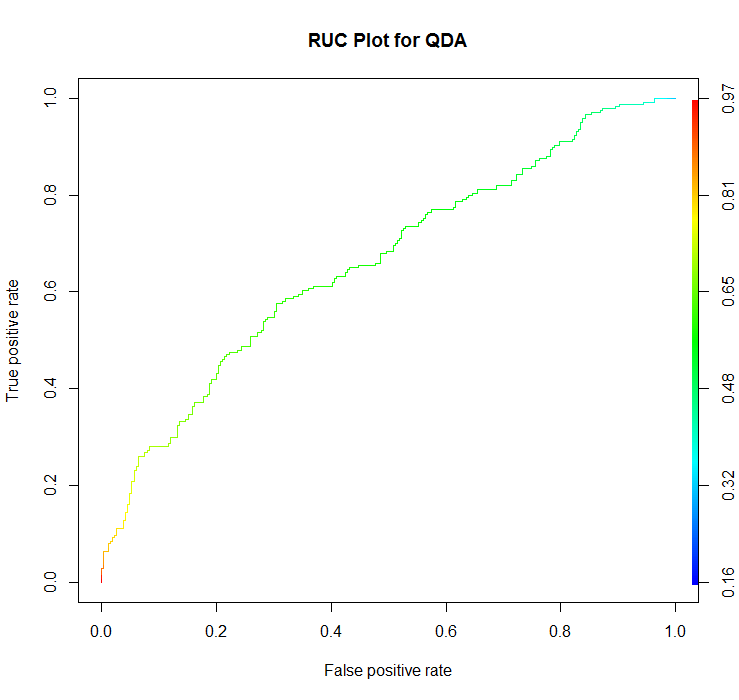
|  |  |  |
| --- | --- | --- |
|  | True switch | True no switch |
| Predicted switch | 218 (TP) | 221 (FP) |
| Predicted no switch | 16 (FN) | 45 (TN) |

AUC is 0.6607



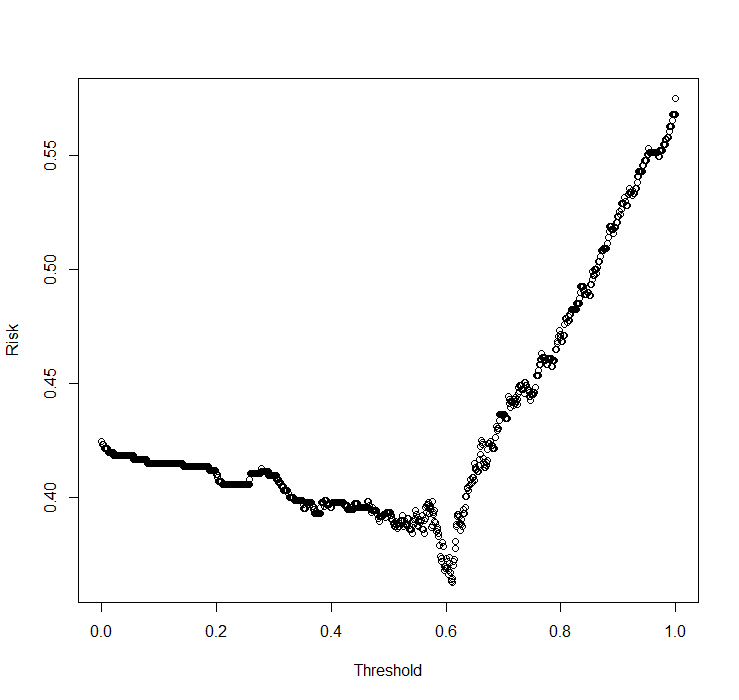
The threshold with the least risk (0.373) was 0.5996. If refitting this into our previous model, our sensitivity would be changed into 0.58 and specificity was 0.70. Compared to the previous model, the predictive ability cannot tell whether exactly increasing or not since the specificity has increased a lot but also sacrifice a lot sensitivity.

1. Quadratic Discriminant Analysis



|  |  |  |
| --- | --- | --- |
|  | True switch | True no switch |
| Predicted switch | 220 (TP) | 225 (FP) |
| Predicted no switch | 14 (FN) | 41(TN) |

AUC is 0.6601



The threshold with the least risk (0.363) was 0.61. If refitting this into our previous model, our sensitivity would be changed into 0.65 and specificity was 0.62. Compared to the previous model, the predictive ability increased since the specificity has increased a lot and almost the same as sensitivity.

1. KNN

When K=1,

|  |  |  |
| --- | --- | --- |
|  | True switch | True no switch |
| Predicted switch | 208 (TP) | 51(FP) |
| Predicted no switch | 26 (FN) | 215(TN) |

AUC is 0.85

According to the high sensitivity and high specificity, this model performed well in both predicting the household has switched wells and in predicting the household has not switched wells.

When K=5,

|  |  |  |
| --- | --- | --- |
|  | True switch | True no switch |
| Predicted switch | 212 (TP) | 78(FP) |
| Predicted no switch | 22 (FN) | 188(TN) |

AUC is 0.81

According to the high sensitivity and high specificity, this model performed well in both predicting the household has switched wells and in predicting the household has not switched wells. But it was a bit worse than K=1 case.

Appendix

# Homework 4

# Read the dataset

setwd("D:/Dropbox (Brown)/Fall2017/PHP2550/HW4")

dat <- read.table("wells.txt", sep = " ", header = T)

# switch is a binary indicator for whether the household switched wells

dat$switch <- as.factor(dat$switch)

# arsenic is the level of arsenic in the well in hundreds of micrograms per liter

# dist is the distance to the nearest safe well in meters

# assoc is whether household members are active in community organizations

dat$assoc <- as.factor(dat$assoc)

# educ is the number of years of education of the head of household.

str(dat)

# Separate into train and test set

train\_dat <- dat[1:2520,]

test\_dat <- dat[2521:3020, ]

### a.

# Test each predictor

summary(glm(switch ~ arsenic, family = binomial(link = logit), data = train\_dat))

summary(glm(switch ~ assoc, family = binomial(link = logit), data = train\_dat))

summary(glm(switch ~ educ, family = binomial(link = logit), data = train\_dat))

summary(glm(switch ~ dist, family = binomial(link = logit), data = train\_dat))

m0 <- glm(switch ~ arsenic + assoc + educ + dist, family = binomial(link = logit), data = train\_dat)

summary(m0)

m1 <- glm(switch ~ -1 + arsenic + educ + dist + assoc, family = binomial(link = logit), data = train\_dat)

summary(m1)

m2 <- glm(switch ~ -1 + arsenic + educ + dist, family = binomial(link = logit), data = train\_dat)

summary(m2)

anova(m1, m2,test = "Chisq")

m3 <- glm(switch ~ -1 + arsenic \* educ + dist, family = binomial(link = logit), data = train\_dat)

summary(m3)

m4 <- glm(switch ~ -1 + arsenic + educ \* dist, family = binomial(link = logit), data = train\_dat)

summary(m4)

m5 <- glm(switch ~ -1 + arsenic \* dist + educ, family = binomial(link = logit), data = train\_dat)

summary(m5) # not significant

m6 <- glm(switch ~ -1 + arsenic + educ \* dist +educ\*arsenic, family = binomial(link = logit), data = train\_dat)

summary(m6)

anova(m2, m6,test = "Chisq")

model <- glm(switch ~ -1 + arsenic + educ \* dist, family = binomial(link = logit), data = train\_dat)

summary(model)

anova(m2,model,test = "Chisq")

# b

p <- function(z){

return(exp(z)/(1+exp(z)))

}

z <- function(arsenic, educ, dist){

return(coef(model)[1]\*arsenic+coef(model)[2]\*educ+coef(model)[3]\*dist+coef(model)[4]\*educ\*dist)

}

# EDUC

range(train\_dat$dist)

x <- seq(0, 340, length.out = 2520)

plot(x, as.numeric(as.character(train\_dat$switch)),xlim=c(0,350), ylim = c(-0.1,1.1), xlab = "Distance",

ylab="probability of switch", main = c("Stratifying by the number of years of education of the head of household."))

range(train\_dat$educ)

educs <- c(0,4,8,12,16)

cols <- c("red", "blue", "green", "orange", "purple")

for (i in seq\_along(educs)){

lines(x, p(z(mean(train\_dat$arsenic),educ = educs[i], x)), col=cols[i])

}

abline(h=0.5)

# ARSENIC

plot(x, as.numeric(as.character(train\_dat$switch)), ylim = c(-0.1,1.1), xlab = "Distance", ylab="probability of switch",

main = c("Stratifying by the level of arsenic in the well in hundreds of micrograms per liter"))

range(train\_dat$arsenic)

arsenics <- c(0.5, 2.5, 4.5, 6.5, 8.5)

for (i in seq\_along(educs)){

lines(x, p(z(arsenics[i],educ = mean(train\_dat$arsenic), x)), col=cols[i])

}

abline(h=0.5)

# DIST

x1 <- seq(0, 9, length.out = 2520)

plot(x1, as.numeric(as.character(train\_dat$switch)), ylim = c(-0.1,1.1), xlab = "Arsenic", ylab="probability of switch",

main = c("Stratifying by the distance to the nearest safe well in meters"))

for (i in seq\_along(educs)){

lines(x, p(z(x1,educ = educs[i], mean(train\_dat$dist))), col=cols[i])

}

abline(h=0.5)

#c

prob <- predict.glm(model, type = "response", newdata = test\_dat)

pred <- ifelse(prob>0.5,1,0)

table(pred, test\_dat$switch)

# Sensitivity

210/(24+210)

sensitivity(confusion.matrix(test\_dat$switch, prob, threshold = 0.5))

# Specificity

58/(208+58)

specificity(confusion.matrix(test\_dat$switch, prob, threshold = 0.5))

# 1- Accuracy

1-(210+58)/(58+24+208+210)

# d

library(pROC)

roc(test\_dat$switch, prob,plot = T)

# e

library(SDMTools)

p\_ts <- sum(dat$switch==1)/dim(dat)[1]

risk <- function(p){

cm <- confusion.matrix(test\_dat$switch, prob, threshold = p)

sen <- sensitivity(cm)

spe <- specificity(cm)

return(p\_ts\*(1-sen)+(1-p\_ts)\*(1-spe))

}

x2 <- seq(0,1, length.out = 1000)

y <- mapply(risk, x2)

plot(x2, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk

min(y)

threshold <-x2[which.min(y)]

threshold

cm0 <- confusion.matrix(test\_dat$switch, pred, threshold = threshold)

sensitivity(cm0)

specificity(cm0)

# f

# LDA

library(MASS)

library(ROCR)

mlda0 <- lda(switch ~ -1 + arsenic + educ \* dist, train\_dat)

mlda0\_fit <- predict(mlda0, test\_dat)

prob\_lda <- predict(mlda0, test\_dat)$posterior[,2]

pred\_lda <- prediction(prob\_lda, test\_dat$switch)

perf\_lda <- performance(pred\_lda, "tpr", "fpr")

plot(perf\_lda,main="RUC Plot for LDA", colorize=TRUE)

table(mlda0\_fit$class, test\_dat$switch)

performance(pred\_lda, "auc")@y.values

p\_ts <- sum(dat$switch==1)/dim(dat)[1]

risk <- function(p){

cm <- confusion.matrix(test\_dat$switch, prob\_lda, threshold = p)

sen <- sensitivity(cm)

spe <- specificity(cm)

return(p\_ts\*(1-sen)+(1-p\_ts)\*(1-spe))

}

x3 <- seq(0,1, length.out = 1000)

y <- mapply(risk, x3)

plot(x3, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk

min(y)

threshold <-x3[which.min(y)]

threshold

cm1 <- confusion.matrix(test\_dat$switch, prob\_lda, threshold = threshold)

sensitivity(cm1)

specificity(cm1)

#QDA

mqda0 <- qda(switch ~ -1 + arsenic + educ \* dist, train\_dat)

mqda0\_fit <- predict(mqda0, test\_dat)

prob\_qda <- predict(mqda0, test\_dat)$posterior[,2]

pred\_qda <- prediction(prob\_qda, test\_dat$switch)

perf\_qda <- performance(pred\_qda, "tpr", "fpr")

plot(perf\_lda,main="RUC Plot for QDA", colorize=TRUE)

table(mqda0\_fit$class, test\_dat$switch)

performance(pred\_qda, "auc")@y.values

p\_ts <- sum(dat$switch==1)/dim(dat)[1]

risk <- function(p){

cm <- confusion.matrix(test\_dat$switch, prob\_qda, threshold = p)

sen <- sensitivity(cm)

spe <- specificity(cm)

return(p\_ts\*(1-sen)+(1-p\_ts)\*(1-spe))

}

x4 <- seq(0,1, length.out = 1000)

y <- mapply(risk, x4)

plot(x4, y, xlab = "Threshold", ylab = "Risk")

# Threshold with least risk

min(y)

threshold <-x4[which.min(y)]

threshold

cm2 <- confusion.matrix(test\_dat$switch, prob\_qda, threshold = threshold)

sensitivity(cm2)

specificity(cm2)

# KNN

mknn0 <- knn(train = train\_dat, test = test\_dat,cl=train\_dat$switch, k=1)

table(mknn0, test\_dat$switch)

multiclass.roc(response=test\_dat$switch, predictor = as.ordered(mknn0))

mknn1 <- knn(train = train\_dat, test = test\_dat,cl=train\_dat$switch, k=5)

table(mknn1, test\_dat$switch)

multiclass.roc(response=test\_dat$switch, predictor = as.ordered(mknn1))