**Problem 2 Section 1**

1. Sensitivity = 626/(626+140) = 0.8172324

Specificity = 160/(74+160) = 0.6837607

Training error rate = mean(pred.glm != Default) = 0.214

1. Test error rate of the full model based on GLM, using my own code with LOOCV method

mean((Default - out.glm)^2) = 0.1667307 # with posterior probability 0.1667307

mean((Default - round(out.glm))^2) = 0.249

1. Test error rate of the full model based on GLM, using cv.glm with LOOCV method

boot::cv.glm(german\_credit, fit.full)$delta[1] = 0.1667307

This is the same result we had in b)

1. Test error rate of the proposed model in problem 1 based on GLM, using my own code with LOOCV method

mean((Default - out.fit7)^2) = 0.1636844 # with posterior probability

mean((Default - round(out.fit7))^2) = 0.242

Test error rate of the proposed model in problem 1 based on GLM, using cv.glm with LOOCV method

boot::cv.glm(german\_credit, fit7)$delta[1] = 0.1636844

Test error rate of the full model based on GLM, using cv.glm with LOOCV method

boot::cv.glm(german\_credit, fit.full)$delta[1] = 0.1667307

# Estimated test error rate is 0.3731118

(not sure what model professor needs by saying #1, so did for the full model and proposed model form problem 1)

1. Test error rate of the full model based on LDA, using my own code with LOOCV method

mean((Default - out.lda)^2) = 0.1664054 # with posterior probability

mean((Default - round(out.lda))^2) = 0.242

Test error rate of the full model based on LDA, using caret package with LOOCV method

mean(Default != pred) = 0.223

# Estimated test error rate is 0.390121

1. Test error rate of the full model based on QDA, using my own code with LOOCV method

mean((Default - out.qda)^2) = 0.2468152 # with posterior probability

mean((Default - round(out.qda))^2) = 0.284

Test error rate of the full model based on QDA, using caret package with LOOCV method

mean(Default != pred) = 0.177

# Estimated test error rate is 0.3477322

1. From the caret package optimum k was found to be K = 77

mean(Default != fit.KNN) = 0.288

# Estimated test error rate is 0.07455013

|  |  |  |
| --- | --- | --- |
| Method | Overall Misclassification rate | LOOCV Test Error Rate |
| GLM | 0.214 | 0.3731118 |
| LDA | 0.223 | 0.390121 |
| QDA | 0.177 | 0.3477322 |
| KNN | 0.288 | 0.07455013 |

Even though KNN has a very low Test error rate, training error rate is high compared to others. Considering other methods, QDA shows the lowest test error. Thus I would recommend QDA.

**Problem 2 Section 2**

### problem 2.

library(caret) # for cross-validation

library(MASS) # for LDA and QDA

library(cvAUC) # for calculating AUC

library(class) # for knn

german\_credit <- read.csv("germancredit.csv", header = T)

attach(german\_credit)

### problem 2. a)

fit.full <- glm(Default ~. , family = binomial, data = german\_credit)

pred <- predict(fit.full, german\_credit, type = "response")

pred.glm <- ifelse(pred > 0.5, 1, 0)

#confusion Matrix

table(Default, pred > 0.5)

#sensitivity

626/(626+140) # [1] 0.8172324

#specificity

160/(74+160) # [1] 0.6837607

#overall misclassification rate

(140+74)/(626+140+74+160) # [1] 0.214

# Error rate based on training data

mean(pred.glm != Default)

# [1] 0.214

### problem 2. b)

fit.full <- glm(Default ~. , family = binomial, data = german\_credit)

out.glm <- NULL

for (i in 1:nrow(german\_credit))

out.glm[i] <- predict(update(fit.full, data = german\_credit[-i,]),

newdata = german\_credit[i,], type = "response")

mean((Default - out.glm)^2) # with posterior probability

# [1] 0.1667307

mean((Default - round(out.glm))^2)

# [1] 0.249

### problem 2. c)

boot::cv.glm(german\_credit, fit.full)$delta[1]

# [1] 0.1667307

### problem 2. d)

fit7 <- glm(Default ~ factor(checkingstatus1) + duration + factor(history) + factor(purpose) + amount +

factor(savings) + installment + factor(status) + factor(others) +

factor(otherplans) + factor(housing) +

factor(foreign), family = binomial, data = german\_credit)

out.fit7 <- NULL

for (i in 1:nrow(german\_credit))

out.fit7[i] <- predict(update(fit7, data = german\_credit[-i,]), newdata = german\_credit[i,], type = "response")

mean((Default - out.fit7)^2) # with posterior probability

# [1] 0.1636844

mean((Default - round(out.fit7))^2)

# [1] 0.242

boot::cv.glm(german\_credit, fit7)$delta[1]

# [1] 0.1636844

boot::cv.glm(german\_credit, fit.full)$delta[1]

# [1] 0.1667307

set.seed(1)

fit.full.GLM.CARET <- train(as.factor(Default) ~ . ,

data = german\_credit,

method ="glm",

trControl = trainControl(method = "LOOCV"))

pred <- as.numeric(predict(fit.full.GLM.CARET, german\_credit)) - 1

mean((Default - pred)^2)

# [1] 0.214

# Estimated test error rate is 0.3731118

### problem 2. e)

out.lda <- NULL

fit.full.LDA <- lda(Default ~. , data = german\_credit)

for (i in 1:nrow(german\_credit))

out.lda[i] <- predict(update(fit.full.LDA, data = german\_credit[-i,]),

newdata = german\_credit[i,], type = "response")$posterior[,2]

mean((Default - out.lda)^2) # with posterior probability

# [1] 0.1664054

mean((Default - round(out.lda))^2)

mean(Default != round(out.lda))

# [1] 0.242

set.seed(1)

fit.full.LDA.CARET <- train(as.factor(Default) ~ . ,

data = german\_credit,

method ="lda",

trControl = trainControl(method = "LOOCV"))

pred <- as.numeric(predict(fit.full.LDA.CARET, german\_credit)) - 1

mean((Default - pred)^2)

# [1] 0.223

# Estimated test error rate is 0.390121

### problem 2. f)

out.qda <- NULL

fit.full.QDA <- qda(Default ~. , data = german\_credit)

for (i in 1:nrow(german\_credit))

tryCatch({

out.qda[i] <- predict(update(fit.full.QDA, data = german\_credit[-i,]),

newdata = german\_credit[i,], type = "response")$posterior[,2]

}, error=function(e){cat(i, "iteration", "ERROR :",conditionMessage(e), "\n")})

out.qda[204] = 0.5

mean((Default - out.qda)^2) # with posterior probability

# [1] 0.2468152

mean((Default - round(out.qda))^2)

mean(Default != round(out.qda))

# [1] 0.284

set.seed(1)

fit.full.QDA.CARET <- train(as.factor(Default) ~ . ,

data = german\_credit,

method ="qda",

trControl = trainControl(method = "LOOCV"))

pred <- as.numeric(predict(fit.full.QDA.CARET, german\_credit)) - 1

mean(Default != pred)

mean((Default - pred)^2)

# [1] 0.177

# Estimated test error rate is 0.3477322

### problem 2. g)

set.seed(1)

fit.full.KNN <- train(as.factor(Default) ~ .,

method = "knn",

tuneLength = 50,

trControl = trainControl(method="LOOCV"),

data = german\_credit)

# > fit.full.KNN

# k-Nearest Neighbors

#

# 1000 samples

# 20 predictor

# 2 classes: '0', '1'

#

# No pre-processing

# Resampling: Leave-One-Out Cross-Validation

# Summary of sample sizes: 999, 999, 999, 999, 999, 999, ...

# Resampling results across tuning parameters:

#

# k Accuracy Kappa

# 5 0.652 0.06451613

# 7 0.681 0.10090192

# 9 0.687 0.10775371

# --------------------------

# 99 0.700 0.00000000

# 101 0.700 0.00000000

# 103 0.700 0.00000000

#

# Accuracy was used to select the optimal model using the largest value.

# The final value used for the model was k = 77.

german\_credit2 <- german\_credit # copy of the data set

indx <- sapply(german\_credit2, is.factor) # factor variables

german\_credit2[indx] <- lapply(german\_credit[indx], function(x) as.numeric(x)) # converting factor to numeric

# fit KNN with optimal k = 77 found with caret

fit.KNN <- knn(german\_credit2[,-1], german\_credit2[,-1], Default, k = 77)

mean(Default != fit.KNN)

# [1] 0.288