#Logistic regression with quality dataset (To decide the insurance claim quality)

#The heart study

# Unit 3, Modeling the Expert

# Video 4

# Read in dataset

quality = read.csv("quality.csv")

# Look at structure

str(quality)

# Table outcome

table(quality$PoorCare)

# Baseline accuracy

98/131

# Install and load caTools package

install.packages("caTools")

library(caTools)

# Randomly split data

set.seed(88)

split = sample.split(quality$PoorCare, SplitRatio = 0.75)

split

# Create training and testing sets

qualityTrain = subset(quality, split == TRUE)

qualityTest = subset(quality, split == FALSE)

# Logistic Regression Model

QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data=qualityTrain, family=binomial)

summary(QualityLog)

# Make predictions on training set

predictTrain = predict(QualityLog, type="response")

# Analyze predictions

summary(predictTrain)

tapply(predictTrain, qualityTrain$PoorCare, mean)

# Video 5

# Confusion matrix for threshold of 0.5

table(qualityTrain$PoorCare, predictTrain > 0.5)

# Sensitivity and specificity

10/25

70/74

# Confusion matrix for threshold of 0.7

table(qualityTrain$PoorCare, predictTrain > 0.7)

# Sensitivity and specificity

8/25

73/74

# Confusion matrix for threshold of 0.2

table(qualityTrain$PoorCare, predictTrain > 0.2)

# Sensitivity and specificity

16/25

54/74

# Video 6

# Install and load ROCR package

install.packages("ROCR")

library(ROCR)

# Prediction function

ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)

# Performance function

ROCRperf = performance(ROCRpred, "tpr", "fpr")

# Plot ROC curve

plot(ROCRperf)

# Add colors

plot(ROCRperf, colorize=TRUE)

# Add threshold labels

plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

# The heart study

# Unit 3, The Framingham Heart Study

# Video 3

# Read in the dataset

framingham = read.csv("framingham.csv")

# Look at structure

str(framingham)

# Load the library caTools

library(caTools)

# Randomly split the data into training and testing sets

set.seed(1000)

split = sample.split(framingham$TenYearCHD, SplitRatio = 0.65)

# Split up the data using subset

train = subset(framingham, split==TRUE)

test = subset(framingham, split==FALSE)

# Logistic Regression Model

framinghamLog = glm(TenYearCHD ~ ., data = train, family=binomial)

summary(framinghamLog)

# Predictions on the test set

predictTest = predict(framinghamLog, type="response", newdata=test)

# Confusion matrix with threshold of 0.5

table(test$TenYearCHD, predictTest > 0.5)

# Accuracy

(1069+11)/(1069+6+187+11)

# Baseline accuracy

(1069+6)/(1069+6+187+11)

# Test set AUC (The area under the ROC curve)

library(ROCR)

ROCRpred = prediction(predictTest, test$TenYearCHD)

ROCRperf = performance(ROCRpred, "tpr", "fpr")

plot(ROCRperf)

plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

as.numeric(performance(ROCRpred, "auc")@y.values)

#for more details on ROC

### http://gim.unmc.edu/dxtests/roc3.htm