Advances Computational Econometrics. Chapter 2: Linear and quadratic classification

Thierry Denoeux

3/29/2022

Exercise 1

Question 1

Loading the data:

```r
credit<-read.csv('/Users/Thierry/Documents/R/Data/Economics/default_credit_card.csv',
                 sep=";",header=TRUE)
```

Redefining variables as factors:

```r
credit$X2<-as.factor(credit$X2)
levels(credit$X2)<-c("M","F")
credit$X3<-as.factor(credit$X3)
credit$X4<-as.factor(credit$X3)
credit$Y<-as.factor(credit$Y)
```

Splitting the data randomly between a training set and a test set:

```r
set.seed(29032022)
n<-nrow(credit)
ntrain<-20000
ntest<-n-ntrain
train<-sample(n,ntrain)
credit.train<-credit[train,]
credit.test<-credit[-train,]
```

Question 2

We first fit the LDA model on the training data, and predict the response for the test data (we exclude the qualitative variables X_2, X_3 and X_4, which cannot be handled by LDA):

```r
library(MASS)
fit.lda<-lda(Y~.,data=credit.train[,-c(2,3,4)])
pred.lda<-predict(fit.lda,newdata=credit.test[,-c(2,3,4)])
```

We compute the confusion matrix and the error rate:

```r
perf.lda<-table(pred.lda$class,credit.test$Y)
print(perf.lda)
```

```
##
##        0    1
##   0 7565 1687
##   1  223  525
```

```
err.lda<-1-sum(diag(perf.lda))/ntest
print(err.lda)
```

```
## [1] 0.191
```

We perform the same operations with QDA:

```
fit.qda<-qda(Y~.,data=credit.train[,-c(2,3,4)])
pred.qda<-predict(fit.qda,newdata=credit.test[,-c(2,3,4)])
perf.qda<-table(pred.qda$class,credit.test$Y)
print(perf.qda)
```

```
##
##        0    1
##   0 3152  389
##   1 4636 1823
```

```
err.qda<-1-sum(diag(perf.qda))/ntest
print(err.qda)
```

```
## [1] 0.5025
```

For naive Bayes, we use function `naive_bayes` from package `naivebayes`. This time, we can put the qualitative variables in the model:

```
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
fit.naive<-naive_bayes(Y~.,data=credit.train)
pred.naive<-predict(fit.naive,newdata=credit.test)
perf.naive<-table(pred.naive,credit.test$Y)
print(perf.naive)
```

```
##
## pred.naive    0    1
##          0 5587  820
##          1 2201 1392
```

```
err.naive<-1-sum(diag(perf.naive))/ntest
print(err.naive)
```

```
## [1] 0.3021
```

Finally, we apply logistic regression:

```
fit.logreg<- glm(Y~.,data=credit.train,family=binomial)
pred.logreg<-predict(fit.logreg,newdata=credit.test,type='response')
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
perf.logreg <-table(credit.test$Y,pred.logreg>0.5)
print(perf.logreg)
```

```
##
```

```
##     FALSE TRUE
##   0  7593  195
##   1  1698  514
```

```
err.logreg <-1-sum(diag(perf.logreg))/ntest
print(err.logreg)
```

```
## [1] 0.1893
```

Comparison of error rates:

```
print(c(err.lda,err.qda,err.naive,err.logreg))
```

```
## [1] 0.1910 0.5025 0.3021 0.1893
```

The linear classifiers perform better on this data set.

Question 3

To generate the ROC curve of the LDA classifier, we need to provide the discriminant variable `pred.lda$x` to function `roc` of package `pROC`:

```
library(pROC)
roc_lda<-roc(credit.test$Y,as.vector(pred.lda$x))
```

We do the same for QDA, and plot the ROC curve on the same graph as LDA:

```
roc_qda<-roc(credit.test$Y,as.vector(pred.qda$posterior[,1]))
```

We generate the ROC curves of the naive Bayes classifier. Again, we need to provide a discriminant variable; here, we provide the posterior probability of class 1:

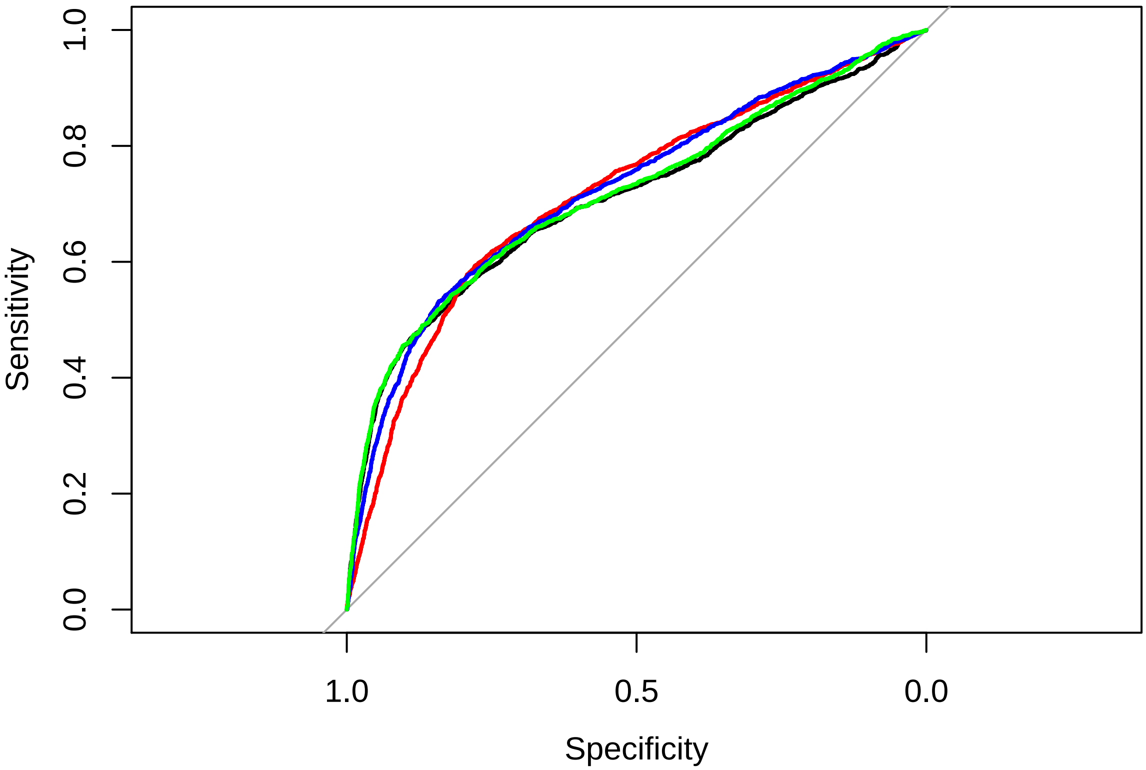
```
pred.nb.prob<-predict(fit.naive,newdata=credit.test,type="prob")
roc_nb<-roc(credit.test$Y,as.vector(pred.nb.prob[,1]))
```

We generate the ROC curve of the logistic regression classifier:

```
roc_logreg<-roc(credit.test$Y,as.vector(pred.logreg))
```

Finally, we plot the four curves on the same graph:

```
plot(roc_lda)
plot(roc_qda,add=TRUE,col='red')
plot(roc_nb,add=TRUE,col='blue')
plot(roc_logreg,add=TRUE,col='green')
```


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity