Predictive Analytics

Solutions to Hands On Exercises

L. Torgo

ltorgo@knoyda.com

KNOYDA, Know Your Data!

Jul, 2019

Hands on LDAs - the Vehicle data set

The data set `Vehicle` is available in package **mlbench**. Load it and explore its help page to grab a minimal understanding of the data and then answer the following questions:

1. Obtain a random split of the data into two sub-sets using the proportion 80%-20%. solution
2. Obtain a linear discriminant using the larger set. solution
3. Obtain the predictions of the obtained model on the smaller set. solution
4. Obtain a confusion matrix of the predictions and calculate the respective accuracy. solution

Solutions to Exercise 1

- Obtain a random split of the data into two sub-sets using the proportion 80%-20%. solution

```
data(Vehicle,package="mlbench")
idx.tr <- sample(1:nrow(Vehicle),as.integer(0.8*nrow(Vehicle)))
tr <- Vehicle[idx.tr,]
ts <- Vehicle[-idx.tr,]
```

Go Back

Solutions to Exercise 2

- Obtain a linear discriminant using the larger set.

```
library(MASS)
model <- lda(Class ~ .,tr)
```

Go Back

Solutions to Exercise 3

- Obtain the predictions of the obtained model on the smaller set.

```
preds <- predict(model,ts)
```

Go Back

Solutions to Exercise 4

- Obtain a confusion matrix of the predictions and calculate the respective accuracy.

```
cm <- table(preds$class,ts$Class)
acc <- sum(diag(cm))/sum(cm)
cat("The accuracy is ",round(acc*100,2),"%.\n")

## The accuracy is  30 %.
```

Hands on SVMs

The file `Wine.Rdata` contains 2 data frames with data about the quality of "green" wines: i) `redWine` and ii) `whiteWine`. Each of these data sets has information on a series of wine tasting sessions to "green" wines (both red and white). For each wine sample several physico-chemical properties of the wine sample together with a quality score assigned by a committee of wine experts (variable `quality`).

1. Obtain and SVM for forecasting the quality of the red variant of "green" wines solution
2. Split the data set in two parts: one with 70% of the samples and the other with the remaining 30%. Obtain an SVM with the first part and apply it to the second. What was the resulting mean absolute error? solution
3. Using the `round()` function, round the predictions obtained in the previous question to the nearest integer. Calculate the error rate of the resulting integers when compared to the true values solution

Solutions to Exercise 1

- Obtain and SVM for forecasting the quality of the red variant of "green" wines

```
load("Wine.Rdata")
library(e1071)
```

```
s <- svm(quality ~ .,redWine)
```

Go back

Solutions to Exercise 2

- Split the data set in two parts: one with 70% of the samples and the other with the remaining 30%. Obtain an SVM with the first part and apply it to the second. What was the resulting mean absolute error?

```
xs <- sample(1:nrow(redWine),
             as.integer(0.7*nrow(redWine)))
train <- redWine[xs,]
test  <- redWine[-xs,]
s2 <- svm(quality ~.,train)
p2 <- predict(s2,test)
mae <- mean(abs(test$quality - p2))
mae

## [1] 0.4358064
```

Go back

Solutions to Exercise 3

- Using the `round()` function, round the predictions obtained in the previous question to the nearest integer. Calculate the error rate of the resulting integers when compared to the true values

```
pi2 <- round(p2)
mc <- table(pi2,test$quality)
mc

##
## pi2   3   4   5   6   7   8
##   5   2   9 165  60   1   0
##   6   0   4  41 114  36   4
##   7   0   0   2  18  23   1
```

Solutions to Exercise 3 (cont.)

```
pi3 <- factor(pi2,levels=levels(factor(test$quality)))
mc2 <- table(pi3,test$quality)
mc2

##
## pi3   3   4   5   6   7   8
##   3   0   0   0   0   0   0
##   4   0   0   0   0   0   0
##   5   2   9 165  60   1   0
##   6   0   4  41 114  36   4
##   7   0   0   2  18  23   1
##   8   0   0   0   0   0   0

err <- 1-sum(diag(mc2))/sum(mc2)
err

## [1] 0.3708333
```

Is this as bad as it looks like?

Go back

Hands on Linear Regression - the Boston data set

The data set `Boston` is available in package **MASS**. Load it and explore its help page to grab a minimal understanding of the data and then answer the following questions:

1. Obtain a random split of the data into two sub-sets using the proportion 70%-30%. solution
2. Obtain a multiple linear regression model using the larger set. solution
3. Check the diagnostic information provided for the model. solution
4. Obtain the predictions of the obtained model on the smaller set. solution
5. Obtain the mean squared error of these predictions and also an error scatter plot. solution

Solutions to Exercise 1

- Obtain a random split of the data into two sub-sets using the proportion 70%-30%. solution

```
data(Boston,package="MASS")
idx.tr <- sample(1:nrow(Boston),as.integer(0.7*nrow(Boston)
tr <- Boston[idx.tr,]
ts <- Boston[-idx.tr,]
```

Go Back

Solutions to Exercise 2

- Obtain a multiple linear regression model using the larger set.

```
model <- lm(medv ~ ., tr)
```

Go Back

Solutions to Exercise 3

- Check the diagnostic information provided for the model.

```
summary(model)


##
## Call:
## lm(formula = medv ~ ., data = tr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.776  -2.371  -0.664   1.906  25.349
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.53e+01   5.93e+00    4.27  2.6e-05 ***
## crim        -6.74e-02   3.91e-02   -1.73  0.08534 .
## zn           4.96e-02   1.50e-02    3.29  0.00109 **
## indus        5.14e-04   6.43e-02    0.01  0.99364
## chas         3.28e+00   9.18e-01    3.57  0.00041 ***
## nox         -1.09e+01   4.07e+00   -2.67  0.00793 **
## rm           4.71e+00   4.96e-01    9.50  < 2e-16 ***
## age         -3.91e-02   1.45e-02   -2.69  0.00749 **
## dis         -1.60e+00   2.22e-01   -7.18  4.4e-12 ***
## rad          2.38e-01   7.48e-02    3.18  0.00162 **
## tax         -1.34e-02   4.15e-03   -3.22  0.00140 **
## ptratio     -8.23e-01   1.40e-01   -5.89  9.4e-09 ***
## black        1.39e-02   3.19e-03    4.34  1.9e-05 ***
## lstat       -3.93e-01   5.46e-02   -7.19  4.1e-12 ***
## ---
```

Solutions to Exercise 4

- Obtain the predictions of the obtained model on the smaller set.

```
preds <- predict(model,ts)
```

Go Back

Solutions to Exercise 5

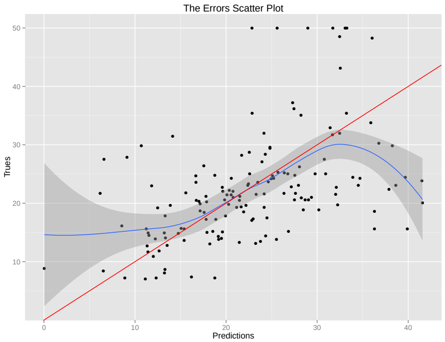
- Obtain the mean squared error of these predictions and also an error scatter plot.

```
mse <- mean((preds-ts$medv)^2)
cat("The mean squared error is ",round(mse*100,2),"\n")

## The mean squared error is  6203
```

Solutions to Exercise 5 (cont.)

- Obtain the mean squared error of these predictions and also an error scatter plot.

```
library(ggplot2)
ggplot(data.frame(Predictions=preds,Trues=ts$medv),aes(x=Predictions,y=Trues)) +
    geom_point() + geom_smooth(method='loess') +
        geom_abline(slope=1, intercept=0,color="red") + ggtitle("The Errors Scatter Plot")
```



Hands on Tree-based Models - the Wines data

File `Wine.Rdata` contains two data frames with data on green wine quality: (i) `redWine` and (ii) `whiteWine`. Each of these data sets contains a series of tests with green wines (red and white). For each of these tests the values of several physicochemical variables together with a quality score assigned by wine experts (column `quality`).

1. Build a regression tree for the white wines data set solution
2. Obtain a graph of the obtained regression tree solution
3. Apply the tree to the data used to obtain the model and calculate the mean squared error of the predictions solution
4. Split the data set in two parts: 70% of the tests and the remaining 30%. Using the larger part to obtain a regression tree and apply it to the other part. Calculate again the mean squared error. Compare with the previous scores and comment. solution

Solutions Exercise 1

- Build a regression tree for the white wines data set

```
load("Wine.Rdata")
library(DMwR2)
```

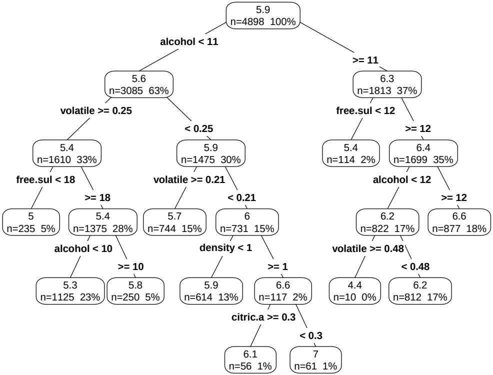
```
ab <- rpartXse(quality ~ .,whiteWine)
```

Go back

Solutions Exercise 2

- Obtain a graph of the obtained regression tree

```
library(rpart.plot)
prp(ab,type=4,extra=101)
```



Solutions Exercise 3

- Apply the tree to the data used to obtain the model and calculate the mean squared error of the predictions

```
prevs <- predict(ab,whiteWine)
mse <- mean((whiteWine$quality - prevs)^2)
mse

## [1] 0.5382
```

Go back

Solutions Exercise 4

- Split the data set in two parts: 70% of the tests and the remaining 30%. Using the larger part to obtain a regression tree and apply it to the other part. Calculate again the mean squared error. Compare with the previous scores and comment.

```
xs <- sample(1:nrow(whiteWine),as.integer(0.7*nrow(whiteWine)))
train <- whiteWine[xs,]
test <- whiteWine[-xs,]
ab2 <- rpartXse(quality ~.,train)
prevs2 <- predict(ab2,test)
mse2 <- mean((test$quality - prevs2)^2)
c(before=mse,now=mse2)


##    before       now
## 0.5382229 0.6037395
```

Go back

Hands on Linear Regression and Random Forests

the Algae data set

Load in the data set `algae` from package **DMwR2** and answer the following questions:

1. How would you obtain a random forest to forecast the value of alga *a4* solution
2. Repeat the previous exercise but now using a linear regression model. Try to simplify the model using the `step()` function. solution
3. Obtain the predictions of the two previous models for the data used to obtain them. Draw a scatterplot comparing these predictions solution
4. The data frame named `test.algae` contains a test set with some extra 140 water samples for which we want predictions. Use the previous two models to obtain predictions for `a4` on these new samples. Check what happened to the test cases with NA's. Fill-in the NA's on the test set and repeat the experiment. solution

Solutions to Exercise 1

- How would you obtain a random forest to forecast the value of alga *a4*

```
library(randomForest)
library(DMwR2)
data(algae)
algae <- algae[-c(62,199),]
algae <- knnImputation(algae)
rf.a4 <- randomForest(a4 ~.,algae[,c(1:11,15)])
```

Go back

Solutions to Exercise 2

- Repeat the previous exercise but now using a linear regression model. Try to simplify the model using the `step()` function.

```
lm.a4 <- lm(a4 ~ .,algae[,c(1:11,15)])
```

```
lm.a4 <- step(lm.a4)
```

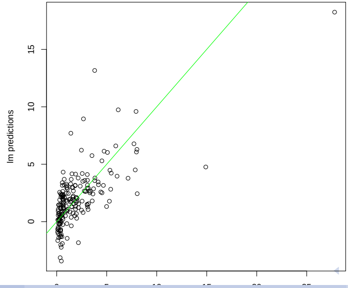
```
lm.a4

##
## Call:
## lm(formula = a4 ~ mxPH + mnO2 + NO3 + NH4 + PO4, data = algae[,
##     c(1:11, 15)])
##
## Coefficients:
## (Intercept)         mxPH         mnO2          NO3          NH4
##   25.155775    -2.564539    -0.307999    -0.466876     0.000932
##         PO4
##    0.009314
```

Go back

Solutions to Exercise 3

- Obtain the predictions of the two previous models for the data used to obtain them. Draw a scatterplot comparing these predictions

```
psrf <- predict(rf.a4,algae)
pslm <- predict(lm.a4,algae)
plot(psrf,pslm,xlab="Random forest predictions",ylab="lm predictions")
abline(0,1,col="green")
```



Solutions to Exercise 4

- The data frame named `test.algae` contains a test set with some extra 140 water samples for which we want predictions. Use the previous two models to obtain predictions for `a4` on these new samples.

```
prevs.rf <- predict(rf.a4,test.algae)
prevs.lm <- predict(lm.a4,test.algae)
summary(prevs.rf)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##  0.09421  0.83539  1.56449  2.12729  2.45076 21.66409       18

summary(prevs.lm) # notice the difference in the number of NA's. Why?

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -2.8201  0.5774  1.5603  2.2330  3.3549 28.6980       6
```

Solutions to Exercise 4 (cont.)

```
test.algae <- knnImputation(test.algae,distData=algae[,1:11])
prevs.rf <- predict(rf.a4,test.algae)
prevs.lm <- predict(lm.a4,test.algae)
```