Exercise Sheet 4 – Data Mining Wirtschaftsinformatik, HTW Berlin

Martin Spott

last revision 05 May 2020

This exercise is about assessing the quality of a classification model. We will use the same data and logistic regression models as in Exercise 3.2.

```r
# load some libraries; install first if necessary
library(ggplot2)
library(ISLR)
library(ROCR)
library(caret)

# load the data set Default in the library ISLR
data(Default)
```

Exercise 4.1

Follow the script below to build and assess a logistic regression model that predicts `default` from `balance`. Remind yourself what the performance measures mean: *Accuracy, Sensitivity (Recall, True Positive Rate, TPR), Specificity (True Negative Rate, TNR)* etc. Compute the *False Positive Rate* and check the help pages for `confusionMatrix` to understand *Balanced Accuracy*.

```r
# build the model
fit <- glm(default~balance, data=Default, family=binomial)

# predict the class probabilities
props <- predict(fit, type="response")

# classify with a threshold
classify <- function(x, threshold) {
  ifelse(x > threshold, 1, 0)
}

# encode the actual class as Yes=1 and No=0
actuals <- ifelse(Default$default == "Yes", 1, 0)

threshold <- 0.5

predicted <- classify(props,threshold)

# specify "1" as the positive class for the performance measures
confusionMatrix(factor(predicted), factor(actuals), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction    0    1
##          0 9625  233
##          1   42  100
##
##                Accuracy : 0.9725
##                  95% CI : (0.9691, 0.9756)
##     No Information Rate : 0.9667
##     P-Value [Acc > NIR] : 0.0004973
##
##                   Kappa : 0.4093
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3003
##             Specificity : 0.9957
##          Pos Pred Value : 0.7042
##          Neg Pred Value : 0.9764
##              Prevalence : 0.0333
##          Detection Rate : 0.0100
##    Detection Prevalence : 0.0142
##       Balanced Accuracy : 0.6480
##
##        'Positive' Class : 1
##
```

Exercise 4.2

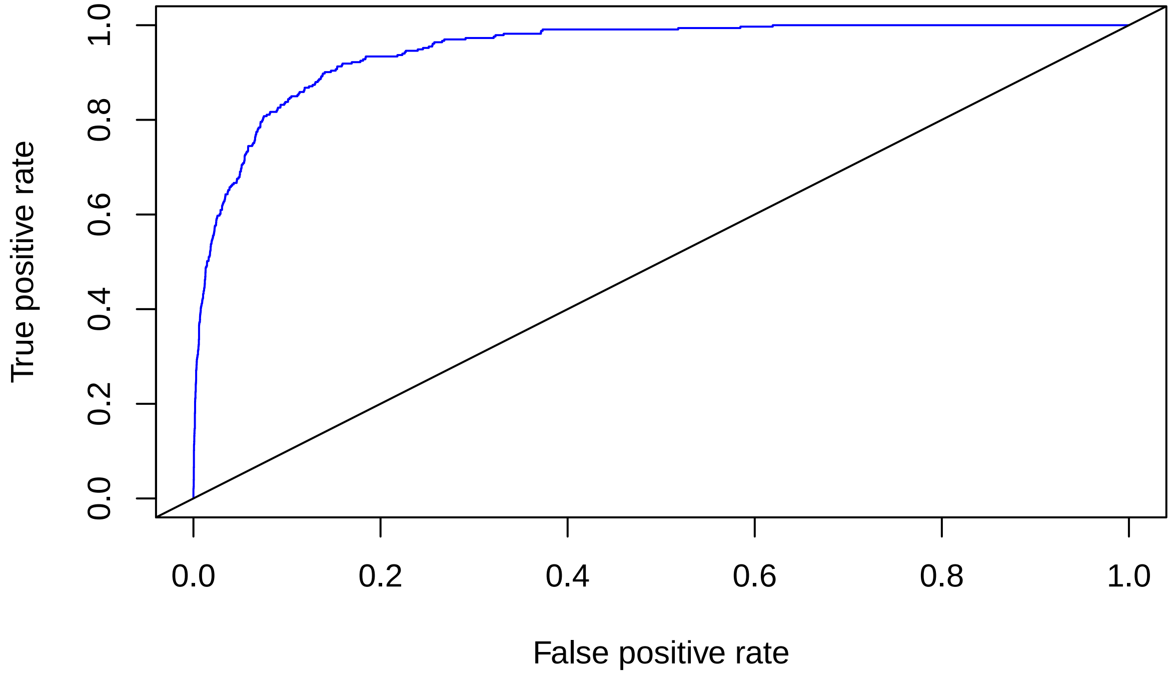
Re-do Exercise 4.1, but change the classification threshold to 0.2 and 0.8. Compare the confusion matrices and the performance measures. Explain the differences in the confusion matrix and figure out, how increasing/decreasing the threshold influences the performance measures (accuracy, true/false positive/negative rate).

Exercise 4.3

Use the following to build the ROC curve and compute the AUC (area under curve). Interpret the ROC curve and the AUC.

```r
# use functions in the library ROCR
# prediction() assumes that the larger value (here 1) is the positive class
pred_rocr <- prediction(props, actuals)
perf_rocr <- performance(pred_rocr, "tpr", "fpr" )

# plot the ROC curve
plot(perf_rocr, col="blue", title="ROC curve")
abline(0,1)
```



```
# The plotted values and associated threshold can be extracted
# as vectors from the performance object
fpr_rocr <- perf_rocr@x.values[[1]]
tpr_rocr <- perf_rocr@y.values[[1]]
threshold_rocr <- perf_rocr@alpha.values[[1]]

# calculate and extract the value for the area under the curve (AUC)
performance(pred_rocr, "auc")@y.values
```

```
## [[1]]
## [1] 0.9479785
```

Exercise 4.4

a) Assign cost values to false positives (FP) and false negatives (FN) as well as to true positives (TP) and true negatives (TN). Measure and compare the costs of the classifiers resulting from Exercises 4.1 and 4.2 as an alternative way to measure the performance. In fact, this is a very important step in practical applications, as financial or social impacts of decisions are more important than pure accuracy.

b) Define a function `cost(prediction, actuals, threshold, cost_fp, cost_fn, cost_tp, cost_tn)` to simplify the comparison in a). Estimate which value for the classification threshold minimises the cost.

Exercise 4.5

Re-do all the steps from Exercises 4.1-4.4, but this time use all attributes in the data set `Default` to predict the column `default`. What is the difference in performance?