ams580\_quiz8

Mustafa Yigit Isik

2023-04-05

# Load packages

if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

#Question 1  
data <- read.csv('/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/quizzes/quiz8/banknote.csv')  
data <- na.omit(data)  
cat('There are', nrow(data), 'observations left.')

## There are 1372 observations left.

data$class <- as.factor(data$class)  
  
set.seed(123)  
training.samples <- data$class %>%  
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data) #1030 obs

## 'data.frame': 1030 obs. of 5 variables:  
## $ variance: num 4.546 3.866 3.457 0.329 4.368 ...  
## $ skewness: num 8.17 -2.64 9.52 -4.46 9.67 ...  
## $ curtosis: num -2.46 1.92 -4.01 4.57 -3.96 ...  
## $ entropy : num -1.462 0.106 -3.594 -0.989 -3.163 ...  
## $ class : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

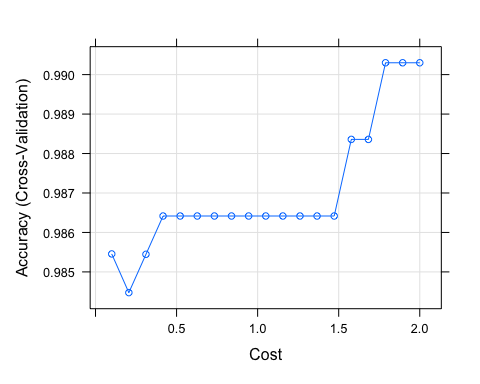
str(test.data) #342 obs

## 'data.frame': 342 obs. of 5 variables:  
## $ variance: num 3.62 3.59 1.9 -1.58 3.4 ...  
## $ skewness: num 8.67 3.01 7.66 10.84 8.73 ...  
## $ curtosis: num -2.807 0.729 0.154 2.546 -2.991 ...  
## $ entropy : num -0.447 0.564 -3.111 -2.936 -0.572 ...  
## $ class : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

#Question 2  
set.seed(123)  
model <- train(  
 class ~., data = train.data, method = "svmLinear",  
 trControl = trainControl("cv", number = 10),  
 )  
# Make predictions on the test data  
predicted.classes <- model %>% predict(test.data)  
# Conf matrix  
confusionMatrix(factor(predicted.classes), factor(test.data$class), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 152 7  
## 1 0 183  
##   
## Accuracy : 0.9795   
## 95% CI : (0.9583, 0.9917)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.9587   
##   
## Mcnemar's Test P-Value : 0.02334   
##   
## Sensitivity : 0.9632   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9560   
## Prevalence : 0.5556   
## Detection Rate : 0.5351   
## Detection Prevalence : 0.5351   
## Balanced Accuracy : 0.9816   
##   
## 'Positive' Class : 1   
##

#Question 3  
set.seed(123)  
model <- train(  
 class ~., data = train.data, method = "svmLinear",  
 trControl = trainControl("cv", number = 10),  
 tuneGrid = expand.grid(C = seq(0.1, 2, length = 19))  
 )  
  
plot(model)



# Print the best tuning parameter sigma and C that  
# maximizes model accuracy  
model$bestTune

## C  
## 17 1.788889

# Make predictions on the test data  
predicted.linear <- model %>% predict(test.data)  
head(predicted.linear)

## [1] 1 1 1 1 1 1  
## Levels: 0 1

# Confusion matrix  
confusionMatrix(factor(predicted.linear), factor(test.data$class), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 151 6  
## 1 1 184  
##   
## Accuracy : 0.9795   
## 95% CI : (0.9583, 0.9917)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9587   
##   
## Mcnemar's Test P-Value : 0.1306   
##   
## Sensitivity : 0.9684   
## Specificity : 0.9934   
## Pos Pred Value : 0.9946   
## Neg Pred Value : 0.9618   
## Prevalence : 0.5556   
## Detection Rate : 0.5380   
## Detection Prevalence : 0.5409   
## Balanced Accuracy : 0.9809   
##   
## 'Positive' Class : 1   
##

#Add pred class label to testing data  
test.data\_pred1 = cbind(test.data, predicted.linear)

#Question 4  
set.seed(123)  
model <- train(  
 class ~., data = train.data, method = "svmRadial",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
 )  
# Print the best tuning parameter sigma and C that  
# maximizes model accuracy  
model$bestTune

## sigma C  
## 2 0.4006328 0.5

# Make predictions on the test data  
predicted.radial <- model %>% predict(test.data)  
head(predicted.radial)

## [1] 1 1 1 1 1 1  
## Levels: 0 1

# Conf matrix  
confusionMatrix(factor(predicted.radial), factor(test.data$class), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 152 0  
## 1 0 190  
##   
## Accuracy : 1   
## 95% CI : (0.9893, 1)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : 1.0000   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 1.0000   
## Prevalence : 0.5556   
## Detection Rate : 0.5556   
## Detection Prevalence : 0.5556   
## Balanced Accuracy : 1.0000   
##   
## 'Positive' Class : 1   
##

#Add pred class label to testing data  
test.data\_pred2 = cbind(test.data, predicted.radial)

#Question 5  
set.seed(123)  
model <- train(  
 class ~., data = train.data, method = "svmPoly",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 4  
 )  
# Print the best tuning parameter sigma and C that  
# maximizes model accuracy  
model$bestTune

## degree scale C  
## 29 2 1 0.25

# Make predictions on the test data  
predicted.poly <- model %>% predict(test.data)  
head(predicted.poly)

## [1] 1 1 1 1 1 1  
## Levels: 0 1

# Conf. matrix  
confusionMatrix(factor(predicted.poly), factor(test.data$class), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 152 0  
## 1 0 190  
##   
## Accuracy : 1   
## 95% CI : (0.9893, 1)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : 1.0000   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 1.0000   
## Prevalence : 0.5556   
## Detection Rate : 0.5556   
## Detection Prevalence : 0.5556   
## Balanced Accuracy : 1.0000   
##   
## 'Positive' Class : 1   
##

#Add pred class label to testing data  
test.data\_pred3 = cbind(test.data, predicted.poly)

#Question 6  
#Out of linear, radial, and polynomial; accuracy results were:  
#linear: 0.9795   
#radial: 1  
#polynomial: 1  
#With these values being pretty close, it's hard to give a clear best case but  
#radial and poly takes the top ever so slightly.

#Question 7  
pred = cbind(test.data\_pred1, test.data\_pred2, test.data\_pred3)  
pred.m = apply(pred,1,function(x) names(which.max(table(x)))) # Majority vote  
pred.m = as.numeric(pred.m)  
confusionMatrix(factor(pred.m), factor(test.data$class), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 152 0  
## 1 0 190  
##   
## Accuracy : 1   
## 95% CI : (0.9893, 1)  
## No Information Rate : 0.5556   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : 1.0000   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 1.0000   
## Prevalence : 0.5556   
## Detection Rate : 0.5556   
## Detection Prevalence : 0.5556   
## Balanced Accuracy : 1.0000   
##   
## 'Positive' Class : 1   
##