Quiz 10

Chaeeun Shin

```r
banknote <- read.csv("banknote.csv")
banknote$class <- as.factor(banknote$class)
banknote <- na.omit(banknote)
cat("There are ", nrow(banknote), "observations left.")
```

```
## There are  1372 observations left.
```

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
set.seed(123)
training_samples <- banknote$class %>%
  createDataPartition(p=0.75,list=FALSE)
train_data <- banknote[training_samples,]
test_data <- banknote[-training_samples,]
nrow(train_data)
```

```
## [1] 1030
```

```r
nrow(test_data)
```

```
## [1] 342
```

```r
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
```

```
##
##     alpha
```

```
set.seed(123)
model1 <- train(class~.,data=train_data,method="svmLinear",trControl=trainControl("cv",number=10))
predicted_class <- model1 %>% predict(test_data)

confusionMatrix(factor(predicted_class),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
##
```

```
set.seed(123)
model2 <- train(class~., data=train_data,method="svmLinear",trControl=trainControl("cv",number=10),tune
```

```
## Warning: model fit failed for Fold01: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold02: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold03: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold04: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold05: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold06: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters
## Warning: model fit failed for Fold07: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold08: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold09: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning: model fit failed for Fold10: C=0.0000 Error in .local(x, ...) :
##   No Support Vectors found. You may want to change your parameters

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.

## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
model2$bestTune
```

```
##           C
## 18 1.789474
```

```
model2 <- train(class~.,data=train_data,method="svmLinear",trControl=trainControl("cv",number=10),tuneG

predicted_class_linear <- model2 %>% predict(test_data)
confusionMatrix(factor(predicted_class_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
##
```

```
predclass <- as.data.frame(predicted_class_linear)
```

```
set.seed(123)
model3 <- train(class~.,data=train_data,method="svmRadial",trControl=trainControl("cv",number=10),tuneLe
model3$bestTune
```

```
##       sigma   C
## 2 0.4006328 0.5
```

```
predicted_class_radial <- model3 %>% predict(test_data)
confusionMatrix(factor(predicted_class_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
##
```

```
predclass <- cbind(predclass,predicted_class_radial)
```

```
set.seed(123)
model4 <- train(class~., data=train_data,method="svmPoly",trControl=trainControl("cv",number=10),tuneLen
model4$bestTune
```

```
##    degree scale    C
## 29      2     1 0.25
```

```
predicted_class_poly <- model4 %>% predict(test_data)
confusionMatrix(factor(predicted_class_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
##
```

```
predclass <- cbind(predclass,predicted_class_poly)
```

6. SVM with radial basis kernel and with polynomial kernel give us the best accuracy. (Accuracy=1)

```
majority_vote <- function(row) {
  counts <- table(row)
  majority_value <- names(counts)[which.max(counts)]
  return(as.numeric(majority_value))
}
predicted_class_mv <- apply(predclass,1,majority_vote)
confusionMatrix(factor(predicted_class_mv),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
##
```