Exercise 11

Nikolaus Czernin

```
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 ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("knitr")
library("e1071")
set.seed(11721138)
```

```
eval_ <- function(y, yhat){
  conf.mat <- table(y, yhat)
  TP <- 0
  FP <- 0
  FN <- 0
  TN <- 0
  try(TP <- conf.mat[2, 2] )
  try(FP <- conf.mat[1, 2] )
  try(FN <- conf.mat[2, 1] )
  try(TN <- conf.mat[1, 1] )
  return (list(TP=TP, FP=FP, FN=FN, TN=TN))
}

# RMSE
RMSE <- function(y, yhat){
  (y - yhat)^2 %>% mean() %>% sqrt()
}

# balanced accuracy: (TPR+TNR)/2
BACC <- function(y, yhat, r=4){
  metrics <- eval_(y, yhat)
  TPR <-  metrics$TP / (metrics$TP + metrics$FN)
  TNR <-  metrics$TN / (metrics$TN + metrics$FP)
  ((TPR + TNR) / 2) %>% round(r)
}
```

Loading & Preprocessing

```
data <- read_delim("bank.csv", delim=";")
```

```
## Rows: 4521 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (10): job, marital, education, default, housing, loan, contact, month, p...
## dbl  (7): age, balance, day, duration, campaign, pdays, previous
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# preprocessing
data <- data %>%
  dplyr::select(-duration) %>%
  mutate(y=ifelse(y=="yes", 1 , 0))


N <- nrow(data)
train_idx <- sample(1:N, N%/%3*2)
train <- data[train_idx,]
test <- data[-train_idx,]
```

a

```
model1 <- svm(y~., data=train, kernel = "radial")
model1
```

```
##
## Call:
## svm(formula = y ~ ., data = train, kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.02380952
##     epsilon:  0.1
##
##
## Number of Support Vectors:  931
```

```
prediction <- (predict(model1, newdata = test, type = "class") > 0) %>% as.numeric()
cf <- table(test$y, prediction) %>% print()
```

```
##    prediction
```

```
##        0    1
##   0    2 1333
##   1    0  172
```

```
paste("Balanced Accuracy:", BACC(test$y, prediction))
```

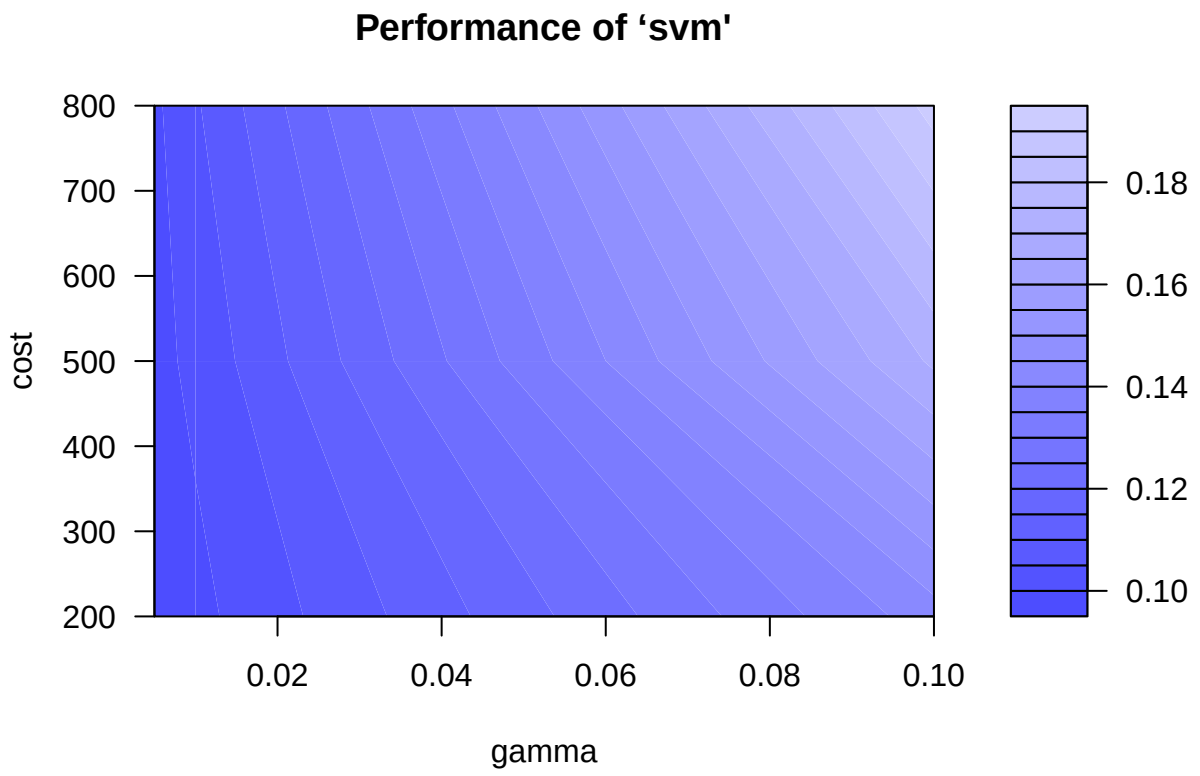
```
## [1] "Balanced Accuracy: 0.5007"
```

b

```
tuned.model <- tune.svm(y~., data=train, kernel = "radial",
                        cost=c(200, 500, 800),
                        gamma=c(0.005, 0.01, 0.1)
                        )
summary(tuned.model)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##  0.005  500
##
## - best performance: 0.09839721
##
## - Detailed performance results:
##   gamma cost      error dispersion
## 1 0.005  200 0.09863136 0.01426583
## 2 0.010  200 0.09856431 0.01473006
## 3 0.100  200 0.14266707 0.01275078
## 4 0.005  500 0.09839721 0.01460372
## 5 0.010  500 0.10125610 0.01476171
## 6 0.100  500 0.17099359 0.01111447
## 7 0.005  800 0.09892070 0.01455593
## 8 0.010  800 0.10436196 0.01545377
## 9 0.100  800 0.19213721 0.01255189
```

```
tuned.model %>% plot()
```



Ive wanted to do a wide grid-search, but the time to compute exploded. It was faster to manually pluck in different parameter values into svm() and note the params that yield the best BACC. Above is me running just 3 parameters each to show the plot, narrowed down to a small value range by trial and error.

c

```
# optimal parameters:
cost <- 500
gamma <- 0.01
best.model <- svm(y~., data=train, kernel = "radial", cost=cost, gamma=gamma)
prediction <- (predict(best.model, newdata = test, type = "class") > 0) %>% as.numeric()
cf <- table(test$y, prediction) %>% print()
```

```
##    prediction
##       0   1
##   0 389 946
##   1  19 153
```

```
paste("Balanced Accuracy:", BACC(test$y, prediction))
```

```
## [1] "Balanced Accuracy: 0.5905"
```

The balanced accuracy increased from using those new parameters. I have paid special attention to keeping the false negatives low. Yet, there are still tons of false positives, which I dont mind as much though.

d

```
tuned_model <- tune(
  svm,
  y ~ .,
  data = train,
  kernel = "radial",
  tunecontrol = tune.control(sampling = "cross", error.fun = BACC),
  ranges = list(
    cost=c(200, 500, 800),
    gamma=c(0.005, 0.01, 0.1)
  ),
  class.weights = c(1, 7) # 7 times as many failures as successes
)
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## ’class.weights’ are set to NULL for regression mode. For classification, use a
## _factor_ for ’y’, or specify the correct ’type’ argument.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## 'class.weights' are set to NULL for regression mode. For classification, use a
## _factor_ for 'y', or specify the correct 'type' argument.
```

```
best.model.2 <- tuned_model$best.model
summary(best.model.2)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = y ~ ., data = train, ranges = list(cost = c(200,
##     500, 800), gamma = c(0.005, 0.01, 0.1)), tunecontrol = tune.control(sampling = "cross",
##     error.fun = BACC), kernel = "radial", class.weights = c(1, 7))
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:
##       gamma:
##     epsilon:  0.1
##
##
## Number of Support Vectors:  698
```

```
prediction <- (predict(best.model.2, newdata = test, type = "class") > 0) %>% as.numeric()
cf <- table(test$y, prediction) %>% print()
```

```
##    prediction
##        1
##   0 1335
##   1  172
```

```r
paste("Balanced Accuracy:", BACC(test$y, prediction))
```

```
## Error in `[.default`(conf.mat, 2, 2) : subscript out of bounds
## Error in `[.default`(conf.mat, 1, 2) : subscript out of bounds
```

```
## [1] "Balanced Accuracy: 0.5"
```

Somehow, the "optimal" model got even worse, it only predicted successes