More Examples with SVM and ANN

STSM2634

2025-05-15

```r
rm(list = ls())

# Load necessary packages
library(Ecdat)
library(e1071)
library(neuralnet)

# Load the data
data(Caschool)
data <- Caschool

str(data)
```

```
## 'data.frame':    420 obs. of  17 variables:
##  $ distcod : int  75119 61499 61549 61457 61523 62042 68536 63834 62331
67306 ...
##  $ county  : Factor w/ 45 levels "Alameda","Butte",..: 1 2 2 2 2 6 29 11 6
25 ...
##  $ district: Factor w/ 409 levels "Ackerman Elementary",..: 362 214 367
132 270 53 152 383 263 94 ...
##  $ grspan  : Factor w/ 2 levels "KK-06","KK-08": 2 2 2 2 2 2 2 2 2 1 ...
##  $ enrltot : int  195 240 1550 243 1335 137 195 888 379 2247 ...
##  $ teachers: num  10.9 11.1 82.9 14 71.5 ...
##  $ calwpct : num  0.51 15.42 55.03 36.48 33.11 ...
##  $ mealpct : num  2.04 47.92 76.32 77.05 78.43 ...
##  $ computer: int  67 101 169 85 171 25 28 66 35 0 ...
##  $ testscr : num  691 661 644 648 641 ...
##  $ compstu : num  0.344 0.421 0.109 0.35 0.128 ...
##  $ expnstu : num  6385 5099 5502 7102 5236 ...
##  $ str     : num  17.9 21.5 18.7 17.4 18.7 ...
##  $ avginc  : num  22.69 9.82 8.98 8.98 9.08 ...
##  $ elpct   : num  0 4.58 30 0 13.86 ...
##  $ readscr : num  692 660 636 652 642 ...
##  $ mathscr : num  690 662 651 644 640 ...
```

```r
# Create a binary target: high_score = 1 if test score > average, else 0
mean_score <- mean(data$testscr, na.rm = TRUE)
data$high_score <- as.numeric(data$testscr > mean_score)

# Select and convert relevant predictors to numeric
data$str <- as.numeric(data$str)         # student-teacher ratio
data$expnstu <- as.numeric(data$expnstu)  # expenditure per student
data$elpct <- as.numeric(data$elpct)   # % English learners
```

```r
data$avginc <- as.numeric(data$avginc)   # avg. household income

# Train-test split
set.seed(100)
index <- sample(1:nrow(data), 0.7 * nrow(data))
train_data <- data[index, ]
test_data <- data[-index, ]

##############################################################################
###
# Train SVM with probability support
train_data$high_score <- as.factor(train_data$high_score)
svm_model <- svm(high_score ~ str + expnstu + elpct + avginc,
                 data = train_data, kernel = "linear", scale = TRUE,
probability = TRUE)

# Predict on test data
svm_preds <- predict(svm_model, newdata = test_data)

##############################################################################
###
# Train ANN
ann_model <- neuralnet(high_score ~ str + expnstu + elpct + avginc,
                       data = train_data,
                       hidden = c(2),
                       linear.output = FALSE,
                       stepmax = 1e7)

# Plot ANN
plot(ann_model)

# Predict on test data
ann_preds <- predict(ann_model, newdata = test_data)

# Threshold ANN outputs
ann_class_preds <- ifelse(ann_preds > 0.5, 1, 0)

##############################################################################
###
# Evaluate model errors
test_data$high_score_num <- as.numeric(as.character(test_data$high_score))

# SVM test error
svm_test_error <- mean(svm_preds != test_data$high_score)
cat("SVM Test Error:", round(svm_test_error, 4), "\n")

## SVM Test Error: 0.2063
```

```r
# ANN test error
ann_test_error <- mean(ann_class_preds != test_data$high_score_num)
cat("ANN Test Error:", round(ann_test_error, 4), "\n")
```

```
## ANN Test Error: 0.5
```

```r
################################################################################
###
# Dummy input (hypothetical school)
dummy <- data.frame(
  str = 18,           # student-teacher ratio
  expnstu = 5000,    # expenditure per student
  elpct = 10,        # % English learners
  avginc = 25         # average household income (in $1000s)
)

# SVM prediction for dummy input
svm_pred_dummy <- predict(svm_model, newdata = dummy, probability = TRUE)
svm_prob <- attr(svm_pred_dummy, "probabilities")

cat(" ◇  SVM Prediction (class):", as.character(svm_pred_dummy), "\n")
```

```
##  🔹  SVM Prediction (class): 1
```

```r
cat(" ◇  SVM Probabilities:\n")
```

```
##  🔹  SVM Probabilities:
```

```r
print(svm_prob)
```

```
##            0         1
## 1 0.05834608 0.9416539
```

```r
# ANN prediction
ann_pred_dummy <- predict(ann_model, newdata = dummy)
ann_class_dummy <- ifelse(ann_pred_dummy > 0.5, 1, 0)

cat(" ◇  ANN Prediction (probability):", round(ann_pred_dummy, 4), "\n")
```

```
##  🔸  ANN Prediction (probability): 0.466 0.5341
```

```r
cat(" ◇  ANN Predicted Class:", ann_class_dummy, "\n")
```

```
##  🔸  ANN Predicted Class: 0 1
```