project2\_svm\_classification

EMVP

2023-05-02

#Q1  
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

data <- read.csv('Titanic.csv')  
data <- subset(data, select = -c(PassengerId,Name,Ticket,Cabin))  
data <- subset(data, is.na(Age) == FALSE)  
data$Survived <- as.factor(data$Survived)  
  
data$Pclass = as.factor(data$Pclass)  
  
data$Age = scale(data$Age)  
data$Fare = scale(data$Fare)  
  
dim(data)[1] # 714 passengers

## [1] 714

set.seed(123)  
training.samples <- data$Survived %>%  
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data) # 536 obs

## 'data.frame': 536 obs. of 8 variables:  
## $ Survived: Factor w/ 2 levels "0","1": 2 2 1 1 2 2 2 1 1 2 ...  
## $ Pclass : Factor w/ 3 levels "1","2","3": 1 3 3 3 2 3 1 3 3 2 ...  
## $ Sex : chr "female" "female" "male" "male" ...  
## $ Age : num [1:536, 1] 0.571 -0.255 0.365 -1.907 -1.081 ...  
## $ SibSp : int 1 0 0 3 1 1 0 0 0 0 ...  
## $ Parch : int 0 0 0 1 0 1 0 0 0 0 ...  
## $ Fare : num [1:536, 1] 0.6914 -0.5059 -0.5035 -0.2574 -0.0874 ...  
## $ Embarked: chr "C" "S" "S" "S" ...

str(test.data) # 178 obs

## 'data.frame': 178 obs. of 8 variables:  
## $ Survived: Factor w/ 2 levels "0","1": 1 2 1 2 1 1 2 1 1 1 ...  
## $ Pclass : Factor w/ 3 levels "1","2","3": 3 1 1 3 3 3 2 1 1 3 ...  
## $ Sex : chr "male" "female" "male" "female" ...  
## $ Age : num [1:178, 1] -0.53 0.365 1.673 -0.186 0.64 ...  
## $ SibSp : int 1 1 0 0 1 1 0 3 1 0 ...  
## $ Parch : int 0 0 0 2 5 0 0 2 0 0 ...  
## $ Fare : num [1:178, 1] -0.5186 0.3478 0.3244 -0.4452 -0.0646 ...  
## $ Embarked: chr "S" "S" "S" "S" ...

#Q2  
set.seed(123)  
model <- train(  
 Survived ~., data = train.data, method = "svmLinear",  
 trControl = trainControl("cv", number = 10),  
 )  
  
predicted.classes <- model %>% predict(test.data)  
head(predicted.classes)

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

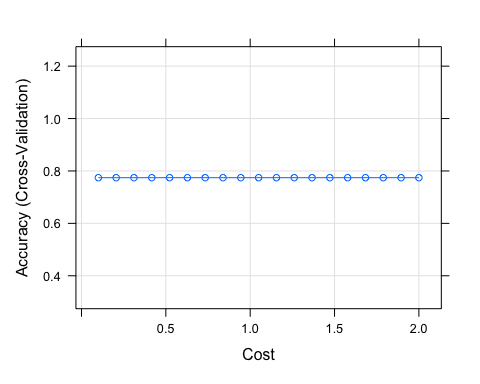
# Confusion matrix  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 93 23  
## 1 13 49  
##   
## Accuracy : 0.7978   
## 95% CI : (0.7312, 0.8541)  
## No Information Rate : 0.5955   
## P-Value [Acc > NIR] : 7.457e-09   
##   
## Kappa : 0.5706   
##   
## Mcnemar's Test P-Value : 0.1336   
##   
## Sensitivity : 0.6806   
## Specificity : 0.8774   
## Pos Pred Value : 0.7903   
## Neg Pred Value : 0.8017   
## Prevalence : 0.4045   
## Detection Rate : 0.2753   
## Detection Prevalence : 0.3483   
## Balanced Accuracy : 0.7790   
##   
## 'Positive' Class : 1   
##

# Accuracy 0.7978  
test.data\_pred = cbind(test.data, predicted.classes)  
str(test.data\_pred)

## 'data.frame': 178 obs. of 9 variables:  
## $ Survived : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 2 1 1 1 ...  
## $ Pclass : Factor w/ 3 levels "1","2","3": 3 1 1 3 3 3 2 1 1 3 ...  
## $ Sex : chr "male" "female" "male" "female" ...  
## $ Age : num [1:178, 1] -0.53 0.365 1.673 -0.186 0.64 ...  
## $ SibSp : int 1 1 0 0 1 1 0 3 1 0 ...  
## $ Parch : int 0 0 0 2 5 0 0 2 0 0 ...  
## $ Fare : num [1:178, 1] -0.5186 0.3478 0.3244 -0.4452 -0.0646 ...  
## $ Embarked : chr "S" "S" "S" "S" ...  
## $ predicted.classes: Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 1 1 1 ...

#Q3  
set.seed(123)  
model <- train(  
 Survived ~., data = train.data, method = "svmLinear",  
 trControl = trainControl("cv", number = 10),  
 tuneGrid = expand.grid(C = seq(0.1, 2, length = 19))  
 )  
  
# Plot model accuracy vs different values of Cost  
plot(model)



model$bestTune

## C  
## 1 0.1

predicted.classes <- model %>% predict(test.data)  
  
# Confusion matrix  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 93 23  
## 1 13 49  
##   
## Accuracy : 0.7978   
## 95% CI : (0.7312, 0.8541)  
## No Information Rate : 0.5955   
## P-Value [Acc > NIR] : 7.457e-09   
##   
## Kappa : 0.5706   
##   
## Mcnemar's Test P-Value : 0.1336   
##   
## Sensitivity : 0.6806   
## Specificity : 0.8774   
## Pos Pred Value : 0.7903   
## Neg Pred Value : 0.8017   
## Prevalence : 0.4045   
## Detection Rate : 0.2753   
## Detection Prevalence : 0.3483   
## Balanced Accuracy : 0.7790   
##   
## 'Positive' Class : 1   
##

# Accuracy 0.7978

#Q4  
set.seed(123)  
model <- train(  
 Survived ~., data = train.data, method = "svmRadial",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
 )  
  
model$bestTune

## sigma C  
## 3 0.1420266 1

predicted.classes <- model %>% predict(test.data)  
  
# Confusion matrix  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 90 17  
## 1 16 55  
##   
## Accuracy : 0.8146   
## 95% CI : (0.7496, 0.8688)  
## No Information Rate : 0.5955   
## P-Value [Acc > NIR] : 3.229e-10   
##   
## Kappa : 0.6143   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.7639   
## Specificity : 0.8491   
## Pos Pred Value : 0.7746   
## Neg Pred Value : 0.8411   
## Prevalence : 0.4045   
## Detection Rate : 0.3090   
## Detection Prevalence : 0.3989   
## Balanced Accuracy : 0.8065   
##   
## 'Positive' Class : 1   
##

# Accuracy 0.8146

#Q5  
set.seed(123)  
model <- train(  
 Survived ~., data = train.data, method = "svmPoly",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 4  
 )  
  
model$bestTune

## degree scale C  
## 25 2 0.1 0.25

predicted.classes <- model %>% predict(test.data)  
  
# Confusion matrix  
confusionMatrix(factor(predicted.classes), factor(test.data$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 93 21  
## 1 13 51  
##   
## Accuracy : 0.809   
## 95% CI : (0.7434, 0.8639)  
## No Information Rate : 0.5955   
## P-Value [Acc > NIR] : 9.523e-10   
##   
## Kappa : 0.5963   
##   
## Mcnemar's Test P-Value : 0.2299   
##   
## Sensitivity : 0.7083   
## Specificity : 0.8774   
## Pos Pred Value : 0.7969   
## Neg Pred Value : 0.8158   
## Prevalence : 0.4045   
## Detection Rate : 0.2865   
## Detection Prevalence : 0.3596   
## Balanced Accuracy : 0.7928   
##   
## 'Positive' Class : 1   
##

# Accuracy 0.809

# Q6  
# Overall accuracy rate:  
# Linear: 0.7978  
# Radial basis kernel: 0.8146 (the best)  
# Polynomial kernel: 0.809