CARET\_Lab

2023-12-08

Package loading

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Loading required package: lattice

library(skimr)

## Warning: package 'skimr' was built under R version 4.3.2

library(ggplot2)  
library(lattice)

Load Data

# attach the iris dataset to the environment  
data(iris)  
# rename the dataset  
dataset <- iris

Task1: Create a Validation/Training Dataset You need to split the loaded dataset into two, 80% of which we will use to train our models and 20% that we will hold back as a validation dataset. Hint: use createDataPartition function

set.seed(123)  
inTraining <- createDataPartition(dataset$Species, p = 0.8, list = FALSE)  
training <- dataset[inTraining, ]  
validation <- dataset[-inTraining, ]

Task2: Summarize Dataset Use skimr library to summarize the dataset

library(skimr)  
skim(training)

Data summary

|  |  |
| --- | --- |
| Name | training |
| Number of rows | 120 |
| Number of columns | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 1 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| Species | 0 | 1 | FALSE | 3 | set: 40, ver: 40, vir: 40 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sepal.Length | 0 | 1 | 5.85 | 0.86 | 4.3 | 5.10 | 5.8 | 6.43 | 7.9 | ▆▇▇▅▃ |
| Sepal.Width | 0 | 1 | 3.05 | 0.41 | 2.0 | 2.80 | 3.0 | 3.30 | 4.1 | ▂▅▇▃▂ |
| Petal.Length | 0 | 1 | 3.77 | 1.79 | 1.1 | 1.58 | 4.3 | 5.10 | 6.9 | ▇▁▆▆▃ |
| Petal.Width | 0 | 1 | 1.21 | 0.78 | 0.1 | 0.30 | 1.3 | 1.83 | 2.5 | ▇▁▆▅▅ |

Task3: split input and output It is the time to separate the input attributes and the output attributes. call the inputs attributes x and the output attribute (or class) y.

x\_train <- training[, 1:4]  
y\_train <- training$Species  
x\_validation <- validation[, 1:4]  
y\_validation <- validation$Species

Task4: Train Control for Validation Test

We will use 10-fold crossvalidation to estimate accuracy.

# Run algorithms using 10-fold cross validation  
control <- trainControl(method="cv", number=10)  
metric <- "Accuracy"

Task5: Model Training Train 5 different algorithms using ‘train’ function:

* Linear Discriminant Analysis (LDA)
* Classification and Regression Trees (CART).
* k-Nearest Neighbors (kNN).
* Support Vector Machines (SVM) with a linear kernel.
* Random Forest (RF)

# Train LDA model  
set.seed(123)  
model\_lda <- train(x\_train, y\_train, method="lda", trControl=control, metric=metric)  
  
# Train CART model  
set.seed(123)  
model\_cart <- train(x\_train, y\_train, method="rpart", trControl=control, metric=metric)  
  
# Train kNN model  
set.seed(123)  
model\_knn <- train(x\_train, y\_train, method="knn", trControl=control, metric=metric)  
  
# Train SVM model  
set.seed(123)  
model\_svm <- train(x\_train, y\_train, method="svmLinear", trControl=control, metric=metric)  
  
# Train RF model  
set.seed(123)  
model\_rf <- train(x\_train, y\_train, method="rf", trControl=control, metric=metric)

Task6: Select the Best Model We now have 5 models and accuracy estimations for each. We need to compare the models to each other and select the most accurate. Use resamples function to complete this task

models <- list(lda = model\_lda, cart = model\_cart, knn = model\_knn, svm = model\_svm, rf = model\_rf)  
resamples(models)

##   
## Call:  
## resamples.default(x = models)  
##   
## Models: lda, cart, knn, svm, rf   
## Number of resamples: 10   
## Performance metrics: Accuracy, Kappa   
## Time estimates for: everything, final model fit

What was the most accurate model?

Task7: Make Prediction (Confusion Matrix) Now we want to get an idea of the accuracy of the best model on our validation set. Use ‘predict’ and confusionMatrix functions to complete this task.

# Predict using the RF model  
predictions <- predict(model\_rf, newdata = x\_validation)  
  
# Confusion Matrix  
confusionMatrix(predictions, y\_validation)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction setosa versicolor virginica  
## setosa 10 0 0  
## versicolor 0 10 2  
## virginica 0 0 8  
##   
## Overall Statistics  
##   
## Accuracy : 0.9333   
## 95% CI : (0.7793, 0.9918)  
## No Information Rate : 0.3333   
## P-Value [Acc > NIR] : 8.747e-12   
##   
## Kappa : 0.9   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: setosa Class: versicolor Class: virginica  
## Sensitivity 1.0000 1.0000 0.8000  
## Specificity 1.0000 0.9000 1.0000  
## Pos Pred Value 1.0000 0.8333 1.0000  
## Neg Pred Value 1.0000 1.0000 0.9091  
## Prevalence 0.3333 0.3333 0.3333  
## Detection Rate 0.3333 0.3333 0.2667  
## Detection Prevalence 0.3333 0.4000 0.2667  
## Balanced Accuracy 1.0000 0.9500 0.9000