

CARET_Lab

Leena Alam

2023-05-16

Package loading

```
library(tidyverse)

## — Attaching packages ————— tidyverse
## 1.3.2 —
## ✓ ggplot2 3.4.0      ✓ purrr 1.0.1
## ✓ tibble 3.2.1       ✓ dplyr 1.1.0
## ✓ tidyr 1.3.0        ✓ stringr 1.5.0
## ✓ readr 2.1.3        ✓ forcats 1.0.0

## Warning: package 'tibble' was built under R version 4.2.3

## — Conflicts —————
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()

library(caret)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
## lift
```

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

```
# Create the training and test datasets
set.seed(100)
```

```
# Step 1: Get row numbers for the training data
trainRowNumbers <- createDataPartition(dataset$Species, p=0.8, list=FALSE)

# Step 2: Create the training dataset
trainData <- dataset[trainRowNumbers,]

# Step 3: Create the test dataset
testData <- dataset[-trainRowNumbers,]
```

Task2: Summarize Dataset Use skimr library to summarize the dataset

```
library(skimr)

## Warning: package 'skimr' was built under R version 4.2.3

skimmed <- skim_to_wide(trainData)

## Warning: 'skim_to_wide' is deprecated.
## Use 'skim()' instead.
## See help("Deprecated")

skimmed
```

Data summary

Name	Piped data
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	p2	p5	p7	p10	hist
Sepal.Length	0	1	5.86	0.82	4.3	5.1	5.8	6.4	7.7	


```

# Set the seed for reproducibility
set.seed(100)

# Train the model using KNN.
KNN_Model = train(Species ~ ., data=trainData, method='knn',
                  trControl = control, metric = metric)

# Set the seed for reproducibility
set.seed(100)

# Train the model using SVM.
SVM_Model = train(Species ~ ., data=trainData, method='svmRadial',
                  trControl = control, metric = metric)

# Set the seed for reproducibility
set.seed(100)

# Train the model using RF.
RF_Model = train(Species ~ ., data=trainData, 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

```

# Compare model performances using resample()
models_compare <- resamples(list(lda = LDA_Model, rpart = CART_Model, knn =
KNN_Model, svmRadial = SVM_Model, rf = RF_Model))

# Summary of the models performances
summary(models_compare)

##
## Call:
## summary.resamples(object = models_compare)
##
## Models: lda, rpart, knn, svmRadial, rf
## Number of resamples: 10
##
## Accuracy
##           Min.   1st Qu.   Median     Mean 3rd Qu.  Max. NA's
## lda      0.9166667 0.9375000 1.0000000 0.9750000      1      1      0
## rpart    0.8333333 0.8541667 0.9166667 0.9250000      1      1      0
## knn      0.9166667 1.0000000 1.0000000 0.9833333      1      1      0
## svmRadial 0.8333333 0.9166667 0.9583333 0.9416667      1      1      0
## rf       0.9166667 0.9166667 1.0000000 0.9666667      1      1      0

```

```
##
## Kappa
##           Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## lda       0.875 0.90625 1.0000 0.9625      1      1      0
## rpart      0.750 0.78125 0.8750 0.8875      1      1      0
## knn        0.875 1.00000 1.0000 0.9750      1      1      0
## svmRadial  0.750 0.87500 0.9375 0.9125      1      1      0
## rf         0.875 0.87500 1.0000 0.9500      1      1      0
```

What was the most accurate model? KNN or LDA

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 on testData (LDA_Model)
predicted <- predict(LDA_Model, testData)

confusionMatrix(reference = testData$Species, data = predicted)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  setosa versicolor virginica
## setosa      10          0          0
## versicolor   0          9          0
## virginica    0          1         10
##
## Overall Statistics
##
##           Accuracy : 0.9667
##           95% CI : (0.8278, 0.9992)
##       No Information Rate : 0.3333
##       P-Value [Acc > NIR] : 2.963e-13
##
##           Kappa : 0.95
##
## Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: setosa Class: versicolor Class: virginica
## Sensitivity           1.0000           0.9000           1.0000
## Specificity           1.0000           1.0000           0.9500
## Pos Pred Value        1.0000           1.0000           0.9091
## Neg Pred Value        1.0000           0.9524           1.0000
## Prevalence            0.3333           0.3333           0.3333
## Detection Rate        0.3333           0.3000           0.3333
## Detection Prevalence  0.3333           0.3000           0.3667
## Balanced Accuracy     1.0000           0.9500           0.9750
```

```
# Predict on testData (CART_Model)
predicted <- predict(CART_Model, testData)

confusionMatrix(reference = testData$Species, data = predicted)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  setosa versicolor virginica
##   setosa      10           0           0
##   versicolor   0           9           3
##   virginica    0           1           7
##
## Overall Statistics
##
##              Accuracy : 0.8667
##              95% CI : (0.6928, 0.9624)
##   No Information Rate : 0.3333
##   P-Value [Acc > NIR] : 2.296e-09
##
##              Kappa : 0.8
##
##   Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##              Class: setosa Class: versicolor Class: virginica
## Sensitivity              1.0000              0.9000              0.7000
## Specificity              1.0000              0.8500              0.9500
## Pos Pred Value           1.0000              0.7500              0.8750
## Neg Pred Value           1.0000              0.9444              0.8636
## Prevalence               0.3333              0.3333              0.3333
## Detection Rate           0.3333              0.3000              0.2333
## Detection Prevalence     0.3333              0.4000              0.2667
## Balanced Accuracy        1.0000              0.8750              0.8250

# Predict on testData (KNN_Model)
predicted <- predict(KNN_Model, testData)

confusionMatrix(reference = testData$Species, data = predicted)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  setosa versicolor virginica
##   setosa      10           0           0
##   versicolor   0           9           2
##   virginica    0           1           8
##
## Overall Statistics
```

```
##
##          Accuracy : 0.9
##          95% CI : (0.7347, 0.9789)
##    No Information Rate : 0.3333
##    P-Value [Acc > NIR] : 1.665e-10
##
##          Kappa : 0.85
##
##  McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##          Class: setosa Class: versicolor Class: virginica
## Sensitivity          1.0000          0.9000          0.8000
## Specificity          1.0000          0.9000          0.9500
## Pos Pred Value       1.0000          0.8182          0.8889
## Neg Pred Value       1.0000          0.9474          0.9048
## Prevalence           0.3333          0.3333          0.3333
## Detection Rate       0.3333          0.3000          0.2667
## Detection Prevalence 0.3333          0.3667          0.3000
## Balanced Accuracy    1.0000          0.9000          0.8750

# Predict on testData (SVM_Model)
predicted <- predict(SVM_Model, testData)

confusionMatrix(reference = testData$Species, data = predicted)

## Confusion Matrix and Statistics
##
##          Reference
## Prediction  setosa versicolor virginica
##   setosa      10         0         0
## versicolor    0         9         1
## virginica     0         1         9
##
## 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          0.9000          0.9000
```

## Specificity	1.0000	0.9500	0.9500
## Pos Pred Value	1.0000	0.9000	0.9000
## Neg Pred Value	1.0000	0.9500	0.9500
## Prevalence	0.3333	0.3333	0.3333
## Detection Rate	0.3333	0.3000	0.3000
## Detection Prevalence	0.3333	0.3333	0.3333
## Balanced Accuracy	1.0000	0.9250	0.9250

Predict on testData (RF_Model)

```
predicted <- predict(RF_Model, testData)
```

```
confusionMatrix(reference = testData$Species, data = predicted)
```

Confusion Matrix and Statistics

##

Reference

Prediction setosa versicolor virginica

setosa 10 0 0

versicolor 0 9 1

virginica 0 1 9

##

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 0.9000 0.9000

Specificity 1.0000 0.9500 0.9500

Pos Pred Value 1.0000 0.9000 0.9000

Neg Pred Value 1.0000 0.9500 0.9500

Prevalence 0.3333 0.3333 0.3333

Detection Rate 0.3333 0.3000 0.3000

Detection Prevalence 0.3333 0.3333 0.3333

Balanced Accuracy 1.0000 0.9250 0.9250