CARET\_Lab

2023-05-28

Package loading

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

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) # for reproducibility  
train\_indices <- createDataPartition(dataset$Species, p = 0.8, list = FALSE)  
train\_data <- dataset[train\_indices, ]  
validation\_data <- dataset[-train\_indices, ]

Task2: Summarize Dataset Use skimr library to summarize the dataset

library(skimr)  
skim(train\_data)

Data summary

|  |  |
| --- | --- |
| Name | train\_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 | 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 seperate the input attributes and the output attributes. call the inputs attributes x and the output attribute (or class) y.

x <- train\_data[, 1:4]  
y <- train\_data[, 5]

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)

set.seed(123) # for reproducibility  
  
# Linear Discriminant Analysis (LDA)  
lda\_model <- train(x, y, method = "lda", trControl = control, metric = metric)  
  
# Classification and Regression Trees (CART)  
cart\_model <- train(x, y, method = "rpart", trControl = control, metric = metric)  
  
# k-Nearest Neighbors (kNN)  
knn\_model <- train(x, y, method = "knn", trControl = control, metric = metric)  
  
# Support Vector Machines (SVM)  
svm\_model <- train(x, y, method = "svmRadial", trControl = control, metric = metric)  
  
# Random Forest (RF)  
rf\_model <- train(x, y, method = "rf", trControl = control, metric = metric)  
  
# Create a list of models  
models <- list(  
 "Linear Discriminant Analysis (LDA)" = lda\_model,  
 "Classification and Regression Trees (CART)" = cart\_model,  
 "k-Nearest Neighbors (kNN)" = knn\_model,  
 "Support Vector Machines (SVM)" = svm\_model,  
 "Random Forest (RF)" = rf\_model  
)

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

set.seed(123) # for reproducibility  
resamples\_results <- resamples(models)  
summary(resamples\_results)

##   
## Call:  
## summary.resamples(object = resamples\_results)  
##   
## Models: Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Random Forest (RF)   
## Number of resamples: 10   
##   
## Accuracy   
## Min. 1st Qu. Median  
## Linear Discriminant Analysis (LDA) 0.9166667 1.0000000 1.0000000  
## Classification and Regression Trees (CART) 0.7500000 0.9166667 0.9166667  
## k-Nearest Neighbors (kNN) 0.9166667 0.9375000 1.0000000  
## Support Vector Machines (SVM) 0.8333333 0.9166667 1.0000000  
## Random Forest (RF) 0.7500000 0.9166667 1.0000000  
## Mean 3rd Qu. Max. NA's  
## Linear Discriminant Analysis (LDA) 0.9833333 1 1 0  
## Classification and Regression Trees (CART) 0.9333333 1 1 0  
## k-Nearest Neighbors (kNN) 0.9750000 1 1 0  
## Support Vector Machines (SVM) 0.9583333 1 1 0  
## Random Forest (RF) 0.9500000 1 1 0  
##   
## Kappa   
## Min. 1st Qu. Median Mean 3rd Qu.  
## Linear Discriminant Analysis (LDA) 0.875 1.00000 1.000 0.9750 1  
## Classification and Regression Trees (CART) 0.625 0.87500 0.875 0.9000 1  
## k-Nearest Neighbors (kNN) 0.875 0.90625 1.000 0.9625 1  
## Support Vector Machines (SVM) 0.750 0.87500 1.000 0.9375 1  
## Random Forest (RF) 0.625 0.87500 1.000 0.9250 1  
## Max. NA's  
## Linear Discriminant Analysis (LDA) 1 0  
## Classification and Regression Trees (CART) 1 0  
## k-Nearest Neighbors (kNN) 1 0  
## Support Vector Machines (SVM) 1 0  
## Random Forest (RF) 1 0

What was the most accurate model? The most accurate model based on the summary of resamples is the Linear Discriminant Analysis (LDA) model. It achieved the highest mean accuracy of 0.9833333 and kappa value of 0.9750.

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.

# Get the best model  
best\_model <- models$`Linear Discriminant Analysis (LDA)`  
  
# Predict on the validation set  
predictions <- predict(best\_model, newdata = validation\_data)  
  
# Create the confusion matrix  
confusionMatrix(predictions, validation\_data$Species)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction setosa versicolor virginica  
## setosa 10 0 0  
## versicolor 0 10 1  
## virginica 0 0 9  
##   
## 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 1.0000 0.9000  
## Specificity 1.0000 0.9500 1.0000  
## Pos Pred Value 1.0000 0.9091 1.0000  
## Neg Pred Value 1.0000 1.0000 0.9524  
## Prevalence 0.3333 0.3333 0.3333  
## Detection Rate 0.3333 0.3333 0.3000  
## Detection Prevalence 0.3333 0.3667 0.3000  
## Balanced Accuracy 1.0000 0.9750 0.9500