Accuracy

Mohamed Nachid

boussiala.nachid@univ-alger3.dz

How To Estimate Model Accuracy in R Using The Caret Package

Estimating Model Accuracy

load the libraries

```
library(caret)
library(klaR)
## load the iris dataset
data(iris)
```

Data Split

define an 80%/20% train/test split of the dataset

```
trainIndex <- createDataPartition(iris$Species, p=0.80, list=FALSE)
data_train <- iris[ trainIndex,]
data_test <- iris[-trainIndex,]
```

train a naive bayes model

```
model_split <- NaiveBayes(Species~., data=data_train)
```

make predictions

```
x_test <- data_test[,1:4]
y_test <- data_test[,5]
predictions <- predict(model_split, x_test)
```

summarize results

```
confusionMatrix(predictions$class, y_test)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         0
##   virginica       0          0        10
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8843, 1)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 4.857e-15
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            1.0000           1.0000
## Specificity                 1.0000            1.0000           1.0000
## Pos Pred Value              1.0000            1.0000           1.0000
## Neg Pred Value              1.0000            1.0000           1.0000
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3333           0.3333
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            1.0000           1.0000
```

Bootstrap

define training control

```
train_control_bt <- trainControl(method="boot", number=100)
```

train the model

```
model_bt <- train(Species~., data=iris, trControl=train_control_bt, method="nb")
```

summarize results

```
print(model_bt)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Bootstrapped (100 reps)
## Summary of sample sizes: 150, 150, 150, 150, 150, 150, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9541182  0.9304025
##    TRUE      0.9561401  0.9334954
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

k-fold Cross Validation

define training control

```
train_control_kfcv <- trainControl(method="cv", number=10)
```

fix the parameters of the algorithm

grid <- expand.grid(.fL=c(0), .usekernel=c(FALSE))

train the model

```
model_kfcv <- train(Species~., data=iris, trControl=train_control_kfcv, method="nb")
```

summarize results

```
print(model_kfcv)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9466667  0.92
##    TRUE      0.9600000  0.94
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

Repeated k-fold Cross Validation

define training control

```
train_control_RKFCV <- trainControl(method="repeatedcv", number=10, repeats=3)
```

train the model

```
model_RKFCV <- train(Species~., data=iris, trControl=train_control_RKFCV, method="nb")
```

summarize results

```
print(model_RKFCV)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results across tuning parameters:
##
```

```
##   usekernel  Accuracy   Kappa
##   FALSE      0.9555556  0.9333333
##    TRUE      0.9600000  0.9400000
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

Leave One Out Cross Validation

define training control

```
train_control_LOOCV <- trainControl(method="LOOCV")
```

train the model

```
model_LOOCV <- train(Species~., data=iris, trControl=train_control_LOOCV, method="nb")
```

summarize results

```
print(model_LOOCV)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 149, 149, 149, 149, 149, 149, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9533333  0.93
##    TRUE      0.9600000  0.94
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```