Week4

Tonya MacDonald

2/24/2021

Overview

One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.

Process

I completed the analysis for the assignment using a random forest algorithm. The model can predict with over 99% accuracy with the 95% confidence interval of (0.9893, 0.9935).

```
library(caret)
library(randomForest)
library(e1071)
```

Download data

Download the data and do some cleanup

```
trainurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

traindt <- read.csv(url(trainurl), na.strings=c("NA","#DIV/0!",""))
testdt <- read.csv(url(testurl), na.strings=c("NA","#DIV/0!",""))

traindt <- traindt[,colSums(is.na(traindt)) == 0]
testdt <- testdt[,colSums(is.na(testdt)) == 0]

#remove first 7 columns
traindt <- traindt[,-c(1:7)]
testdt <- testdt[,-c(1:7)]
```

Partition data

```
#split data into training and testing sets
trainPartitions <- createDataPartition(traindt$classe, p = 0.6, list = FALSE)
inTrain <- traindt[trainPartitions,]
inTest <- traindt[-trainPartitions,]

# create factors
inTest$classe <- factor(inTest$classe)
```

Create model and train with the training partition

```
modelFilename <- "model.RData"
if (!file.exists(modelFilename)){
  model <- train(classe ~ .
                 , data = inTrain
                 , method = "rf" #random forest
                 , metric = "Accuracy"
                 , preProcess=c("center", "scale")
                 , trControl=trainControl(method = "cv"
                                          , number = 4
                                          , p= 0.60
                                          , allowParallel = TRUE
                                          )
                 )

save(model, file = "model.RData")
} else {
  load(file = modelFilename, verbose = TRUE)
}
```

```
## Loading objects:
##   model
```

Check the model with the testing data partition

```
# create predictions based off testing partition
predictions <- predict(model, newdata=inTest)

# display analysis
confusionMatrix(predictions, inTest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2231    6    0    0    0
##          B    1 1505    3    0    0
##          C    0    7 1364    4    1
##          D    0    0    1 1282    3
##          E    0    0    0    0 1438
##
## Overall Statistics
##
##                Accuracy : 0.9967
##                  95% CI : (0.9951, 0.9978)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9958
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9996   0.9914   0.9971   0.9969   0.9972
## Specificity            0.9989   0.9994   0.9981   0.9994   1.0000
## Pos Pred Value         0.9973   0.9973   0.9913   0.9969   1.0000
## Neg Pred Value         0.9998   0.9979   0.9994   0.9994   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1918   0.1738   0.1634   0.1833
## Detection Prevalence   0.2851   0.1923   0.1754   0.1639   0.1833
## Balanced Accuracy      0.9992   0.9954   0.9976   0.9981   0.9986
```

```
# display the variable importance
varImp(model)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 52)
##
##                      Overall
## roll_belt             100.00
## pitch_forearm          58.75
## yaw_belt               51.17
## magnet_dumbbell_z      42.30
## pitch_belt             41.99
## magnet_dumbbell_y      41.76
## roll_forearm           41.67
## accel_dumbbell_y       22.15
## accel_forearm_x        17.75
## magnet_dumbbell_x      17.12
## roll_dumbbell          16.25
## magnet_belt_z          15.22
## accel_dumbbell_z       14.12
## accel_belt_z           13.39
## magnet_forearm_z       13.13
## total_accel_dumbbell   11.84
## magnet_belt_y          11.42
## yaw_arm                11.11
## gyros_belt_z           10.74
## magnet_belt_x          10.48
```