Practical Machine Learning

Vanja Čotić Poturić

2022-09-06

Summary

In this project, we use data of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.The goal of this project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set.

Data

```
## Warning: package 'caret' was built under R version 4.1.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 4.1.3
## Loading required package: lattice
## Warning: package 'RColorBrewer' was built under R version 4.1.3
## Warning: package 'randomForest' was built under R version 4.1.3
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
## Warning: package 'rattle' was built under R version 4.1.3
## Loading required package: tibble
## Warning: package 'tibble' was built under R version 4.1.3
## Loading required package: bitops
## Warning: package 'bitops' was built under R version 4.1.1
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
##
## Attaching package: 'rattle'
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```
##   accel_forearm_z magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## 1            -215              -17              654              476      A
## 2            -216              -18              661              473      A
## 3            -213              -18              658              469      A
## 4            -214              -16              658              469      A
## 5            -214              -17              655              473      A
## 6            -215               -9              660              478      A
```

```
## [1] "A" "B" "C" "D" "E"
```

```
## [1] "carlitos" "pedro"    "adelmo"   "charles"  "eurico"   "jeremy"
```

Data cleaning

```
NA_Count = sapply(1:dim(training)[2],function(x)sum(is.na(training[,x])))
NA_list = which(NA_Count>0)
training = training[,-NA_list]
training = training[,-c(1:7)]
testing = testing[,-NA_list]
testing = testing[,-c(1:7)]
dim(testing)
```

```
## [1] 20 53
```

```
dim(training)
```

```
## [1] 19622    53
```

```
table(training$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Cross Validation

LDA

```
set.seed(1234)
lda_model<-train(classe~ .,method="lda", data=training)
predict_lda<-predict(lda_model,training)
confusionMatrix(predict_lda,as.factor(training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 4568  586  341  191  133
##          B  121 2429  333  130  611
##          C  444  455 2254  379  323
##          D  429  148  411 2383  344
##          E   18  179   83  133 2196
##
## Overall Statistics
##
```

```
##                Accuracy : 0.7048
##                  95% CI : (0.6984, 0.7112)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6264
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8186   0.6397   0.6587   0.7410   0.6088
## Specificity            0.9109   0.9245   0.9012   0.9188   0.9742
## Pos Pred Value         0.7850   0.6703   0.5847   0.6415   0.8417
## Neg Pred Value         0.9267   0.9145   0.9259   0.9476   0.9171
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2328   0.1238   0.1149   0.1214   0.1119
## Detection Prevalence   0.2966   0.1847   0.1965   0.1893   0.1330
## Balanced Accuracy      0.8648   0.7821   0.7799   0.8299   0.7915
```

Decision Tree

```
library(rattle)
set.seed(1357)
dt_model <- train(classe ~., method = "rpart", data = training)
predict_dt<-predict(dt_model,training)
confusionMatrix(predict_dt,as.factor(training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5080 1581 1587 1449  524
##          B   81 1286  108  568  486
##          C  405  930 1727 1199  966
##          D    0    0    0    0    0
##          E   14    0    0    0 1631
##
## Overall Statistics
##
##                Accuracy : 0.4956
##                  95% CI : (0.4885, 0.5026)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3407
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9104  0.33869  0.50468   0.0000  0.45218
```

```
## Specificity            0.6339  0.92145  0.78395   1.0000  0.99913
## Pos Pred Value         0.4970  0.50850  0.33040      NaN  0.99149
## Neg Pred Value         0.9468  0.85310  0.88225   0.8361  0.89008
## Prevalence             0.2844  0.19351  0.17440   0.1639  0.18382
## Detection Rate         0.2589  0.06554  0.08801   0.0000  0.08312
## Detection Prevalence   0.5209  0.12889  0.26638   0.0000  0.08383
## Balanced Accuracy      0.7721  0.63007  0.64431   0.5000  0.72565
```

Random Forest

```
set.seed(2468)
rf_model <- train(classe ~., method = "rf", data = training)
predict_rf<-predict(rf_model,training)
confusionMatrix(predict_rf,as.factor(training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5580    0    0    0    0
##          B    0 3797    0    0    0
##          C    0    0 3422    0    0
##          D    0    0    0 3216    0
##          E    0    0    0    0 3607
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

The Random Forest model is selected because it has the largest accuracy.

```
predict <- predict(rf_model, testing)
predict
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```