Assigment

Alberto Perea

7 3 2021

Introduction

In this assignment data from devices such as awbone Up, Nike FuelBand, and Fitbit will be used to predict the manner in which participants did the exercise. The participants were asked to serfdom barbell lifts correctly and incorrectly in 5 different ways. The participants were asked to do Dumbbell Biceps Curl, the 5 different ways they did it is described as follows:

- Class A : Exactly according to the specification.
- Class B : Throwing the elbows to the front.
- Class C : Lifting the dumbbell only halfway.
- Class D : Lowering the dumbbell only halfway.
- Class E : Throwing the hips to the front.

Data

First the data need to be read and cleaned.

We will first read the data, and explore for NAs and left out the variables that are not needed and the one that has too much NAs

```r
training_csv <- read.csv("pml-training.csv", na.strings = c("NA","#DIV/0!", ""))
validation_csv <- read.csv("pml-testing.csv", na.strings = c("NA","#DIV/0!", ""))
idx <- which(colSums(is.na(training_csv))==0) #Extract columns where there are NO NAs
dim(training_csv[,-idx])[2]
```

```
## [1] 100
```

There are 100 variables that contain NAs which we will left out due that this gives no useful information, at the same time we are going to left out the index, timestamps, names and windows variables. This will make our data set cleaner and we will only be using data that we need.

```r
training_df <- training_csv[,idx]
training_df <- training_df[,-c(1:7)]
validation_df <- validation_csv[,idx]
validation_df <- validation_df[,-c(1:7)] #Both data sets need to have the same columns
```

Data partition Train and Test sets

A partition will be created on the training data set, and two new data sets will be created, training (70% of original data set) and testing (30% of original data set), using as outcome the variable classe.

```r
in_train <- createDataPartition(training_df$classe, p = 0.7, list = FALSE)
training <- training_df[in_train,]
testing <- training_df[-in_train,]
```

Model Selection and Cross Validation

The next 3 models will be used: * Random Forest * Decision Tree * Generalized Boosted Model

First Cross validation will be made, it will be a cross validation with 5 folds

```r
set.seed(1212)
control_rf <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
control_gbm <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
```

Random Forest

Fit random forest model using our training data set

```r
modrf <- train(classe~., data = training, method = "rf", trControl = control_rf)
predrf <- predict(modrf, newdata = testing)
cfmodrf <- confusionMatrix(predrf, as.factor(testing$classe))
```

```r
cfmodrf$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    7    0    0    0
##          B    0 1132    8    0    0
##          C    0    0 1018   26    0
##          D    0    0    0  937    5
##          E    0    0    0    1 1077
```

```r
cfmodrf$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9920136      0.9898961      0.9893938      0.9941262      0.2844520
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

Decision Tree

Fit decision tree model using our training data set

```r
moddt <- rpart(classe~., data = training, method = "class")
preddt <- predict(moddt, newdata = testing, type = "class")
cfmoddt <- confusionMatrix(preddt, as.factor(testing$classe))
```

```r
cfmoddt$table
```

```
##             Reference
## Prediction    A    B    C    D    E
##          A 1541  261   76   83   46
##          B   29  629   80   43  111
##          C   41  130  783  159  129
##          D   40   83   61  635  100
##          E   23   36   26   44  696
```

```r
cfmoddt$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   7.279524e-01   6.536665e-01   7.163866e-01   7.392918e-01   2.844520e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   4.681994e-83
```

Generalized Boosted Model

Fit generalized boosted model using our training data set

```r
modgbm <- train(classe~., data = training, method = "gbm", trControl = control_gbm)
predgbm <- predict(modgbm, newdata = testing)
cfmodgbm <- confusionMatrix(predgbm, as.factor(testing$classe))
```

```r
cfmodgbm$table
```

```
##             Reference
## Prediction    A    B    C    D    E
##          A 1663   54    0    0    1
##          B    5 1056   28    3   12
##          C    2   25  991   32   11
##          D    3    1    5  917   18
##          E    1    3    2   12 1040
```

```r
cfmodgbm$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   9.629567e-01   9.531032e-01   9.578113e-01   9.676363e-01   2.844520e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   6.586448e-13
```

Conclusion

As it can be seen from the models tested before, random forest is the one that achieve the most accurate results, this model will be used for the test the 20 predictions in the validations set

```
pred_val <- predict(modrf, newdata = validation_df)
pred_val
```

```
##  [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
```