Weight lifting analysis

S. Menon

Feb 04, 2018

1 Background

In this project, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

A training sample was given to build the model and test file to predict the outcome Source of data http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

2 Upload Data

Data is uploaded, and there are about 160 variables. A lot of the columns are NA or have errors. Hence all columns invalid data was ignored.

```
trainData <- read.csv(file="C:/Users/smenon/Documents/Coursera/Practical ML/pml-training.csv",
                      na.strings=c('#DIV/0', '', 'NA') ,stringsAsFactors = F,header = TRUE)
testData <- read.csv(file="C:/Users/smenon/Documents/Coursera/Practical ML/pml-testing.csv",
                     na.strings=c('#DIV/0', '', 'NA') ,stringsAsFactors = F,header = TRUE)
dim(trainData)
```

```
## [1] 19622   160
```

```
dim(testData)
```

```
## [1]  20 160
```

```
noNAtrainData = trainData[ , colSums(is.na(trainData)) == 0]
dim(trainData)
```

```
## [1] 19622   160
```

```
#str(trainData)
```

3 Model Build

Next step is to build the model. First step is to split the data into train and test groups

3.1 Create Test and hold-out samples

```
inTrain <- createDataPartition(y=noNAtrainData$classe,p=0.7,list=FALSE)
training <- noNAtrainData[inTrain,]
```

```
testing <- noNAtrainData[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 13737    60
```

```
## [1] 5885   60
```

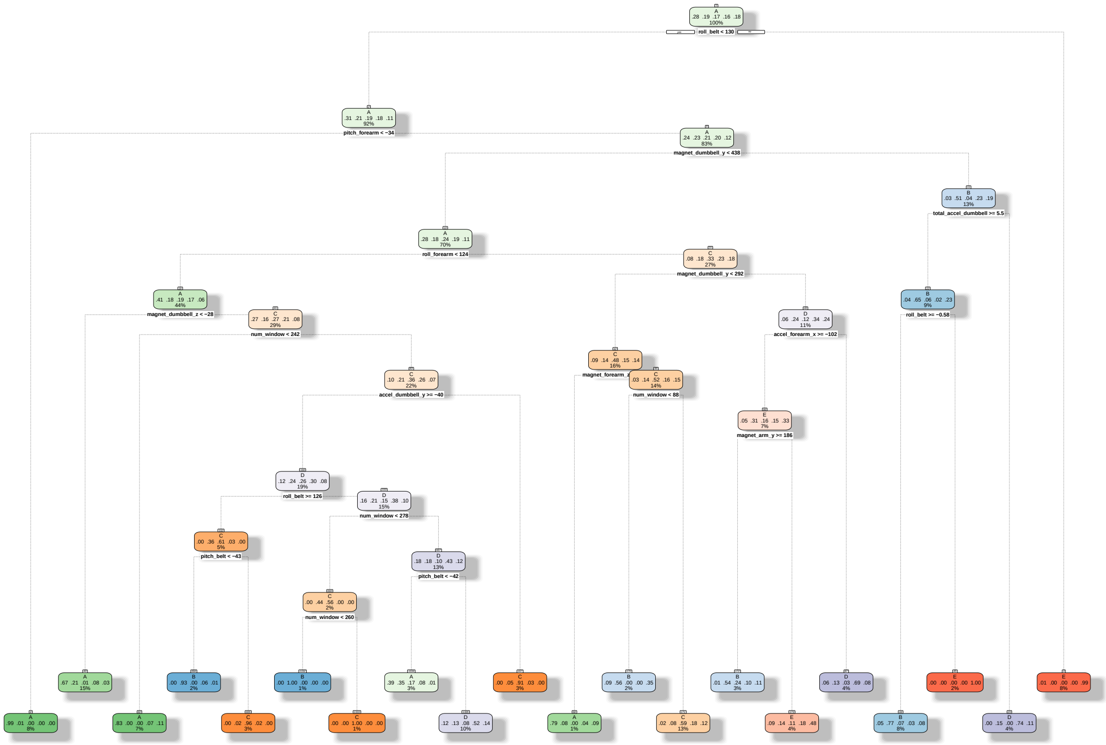
3.2 Build Segmentation Tree

```
fit <- rpart(as.factor(classe) ~ as.factor(user_name) +
             new_window+num_window+roll_belt+
             pitch_belt+yaw_belt+total_accel_belt+gyros_belt_x+
             gyros_belt_y+gyros_belt_z+accel_belt_x+accel_belt_y+
             accel_belt_z+magnet_belt_x+magnet_belt_y+magnet_belt_z+
             roll_arm+pitch_arm+yaw_arm+total_accel_arm+
             gyros_arm_x+gyros_arm_y+gyros_arm_z+accel_arm_x+
             accel_arm_y+accel_arm_z+magnet_arm_x+magnet_arm_y+
             magnet_arm_z+roll_dumbbell+pitch_dumbbell+yaw_dumbbell+
             total_accel_dumbbell+gyros_dumbbell_x+gyros_dumbbell_y+gyros_dumbbell_z+
             accel_dumbbell_x+accel_dumbbell_y+accel_dumbbell_z+magnet_dumbbell_x+
             magnet_dumbbell_y+magnet_dumbbell_z+roll_forearm+pitch_forearm+
             yaw_forearm+total_accel_forearm+gyros_forearm_x+gyros_forearm_y+
             gyros_forearm_z+accel_forearm_x+accel_forearm_y+accel_forearm_z+
             magnet_forearm_x+magnet_forearm_y+magnet_forearm_z,
             data=training, method="class")
```

3.3 Plot Segmentation Tree

```
fancyRpartPlot(fit)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2018−Feb−04 21:01:55 i52279

3.4 Validate model using the hold-out sample

The table below shows comparison of predicted value and actual value of the holdout sample.

```
pred <- predict(fit, newdata = testing,type="class")
testing$predRight <- pred==testing$classe
table(pred,testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1503  250   42   94   72
##    B   47  647   70   41  103
##    C   16   72  836  138   90
##    D   94  133   53  650  125
##    E   14   37   25   41  692
```

Model is able to predict 4160 of 5885 (or 71%) of the cases correctly.

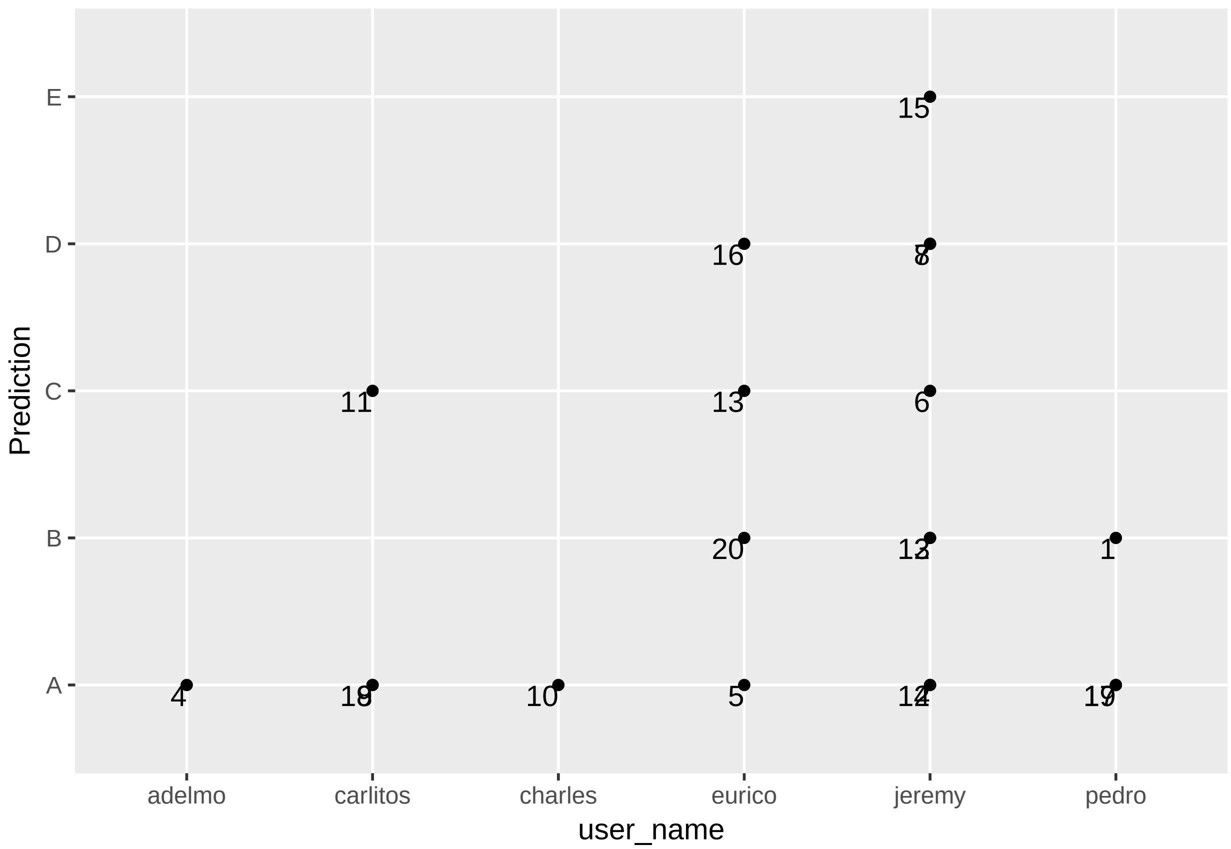
4 Use the segmentation model to predict outcome of test sample provided

```
testData$Prediction <- predict(fit, newdata = testData,type="class")
table(testData$user_name,testData$Prediction)
```

```
##
##             A B C D E
##   adelmo    1 0 0 0 0
##   carlitos  2 0 1 0 0
##   charles   1 0 0 0 0
##   eurico    1 1 1 1 0
##   jeremy    2 2 1 2 1
##   pedro     2 1 0 0 0
```

The table above below shows summary of predicted excercise type by user.

```
ggplot(testData,aes(user_name,Prediction)) + geom_point()+geom_text(aes(label=problem_id),hjust=1,vjust=
```



The chart above provides similar view, with problem_id.