Practical Machine Learning Course Project

Than Win

07/28/2015

Executive Summary

This paper will demonstrate the prediction on the manner of the participents by using carent package taught in the Practical Machine Learning class at Cousera.

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Getting Data

To be able to avoid multiple downloading, the data is saved as rds on local disk.

```
#library(curl)
#trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
#testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

#trainData <- read.csv(curl(trainURL), na.strings=c("NA","#DIV/0!",""))
#testData <- read.csv(curl(testURL), na.strings=c("NA","#DIV/0!",""))

#saveRDS(trainData, file='trainData.rds')
#saveRDS(testData, file='testData.rds')

trainData <- readRDS("/home/ict/trainData.rds")
testData <- readRDS("/home/ict/testData.rds")
```

Explore the data

```
#head(trainData)
dim(trainData)
```

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

```
#str(trainData)

#head(testData)
dim(testData)
```

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

```
#str(trainData$classe)
```

Cleaning the data

```
#remove NA
trainData <- trainData[, colSums(is.na(trainData)) == 0]

#remove unrelevant vars
rm = c('X', 'user_name', 'raw_timestamp_part_1', 'raw_timestamp_part_2', 'cvtd_timestamp', 'new_window'

trainData <- trainData [, -which(names(trainData) %in% rm)]
```

Screening the data

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
#remove Non Zero Value
nzv <- nearZeroVar(trainData, saveMetrics=TRUE)
trainData <- trainData[, nzv [, 'nzv'] == 0]

#find correlation between vars
corMat <- cor(na.omit(trainData[sapply(trainData, is.numeric)]))
dim(corMat)
```

```
## [1] 52 52
```

```
#remove cutoff
nocor <- findCorrelation(corMat, cutoff=0.9, verbose=TRUE)
```

```
## Compare row 10  and column  1 with corr  0.992
##   Means:  0.27 vs 0.168 so flagging column 10
## Compare row 1  and column  9 with corr  0.925
##   Means:  0.25 vs 0.164 so flagging column 1
## Compare row 9  and column  4 with corr  0.928
##   Means:  0.233 vs 0.161 so flagging column 9
## Compare row 8  and column  2 with corr  0.966
##   Means:  0.245 vs 0.157 so flagging column 8
```

```
## Compare row 19  and column  18 with corr  0.918
##   Means:  0.091 vs 0.158 so flagging column 18
## Compare row 46  and column  31 with corr  0.914
##   Means:  0.101 vs 0.161 so flagging column 31
## Compare row 46  and column  33 with corr  0.933
##   Means:  0.083 vs 0.164 so flagging column 33
## All correlations <= 0.9
```

```
trainData <- trainData[,-nocor]
```

Split data for training and cross validation

```
#trainData is further divided into trainDF and validateDF
train <- createDataPartition(y=trainData$classe, p=0.7, list=FALSE)
trainDF <- trainData[train,]
validateDF <- trainData[-train,]
dim(trainDF)
```

```
## [1] 13737    46
```

```
dim(validateDF)
```

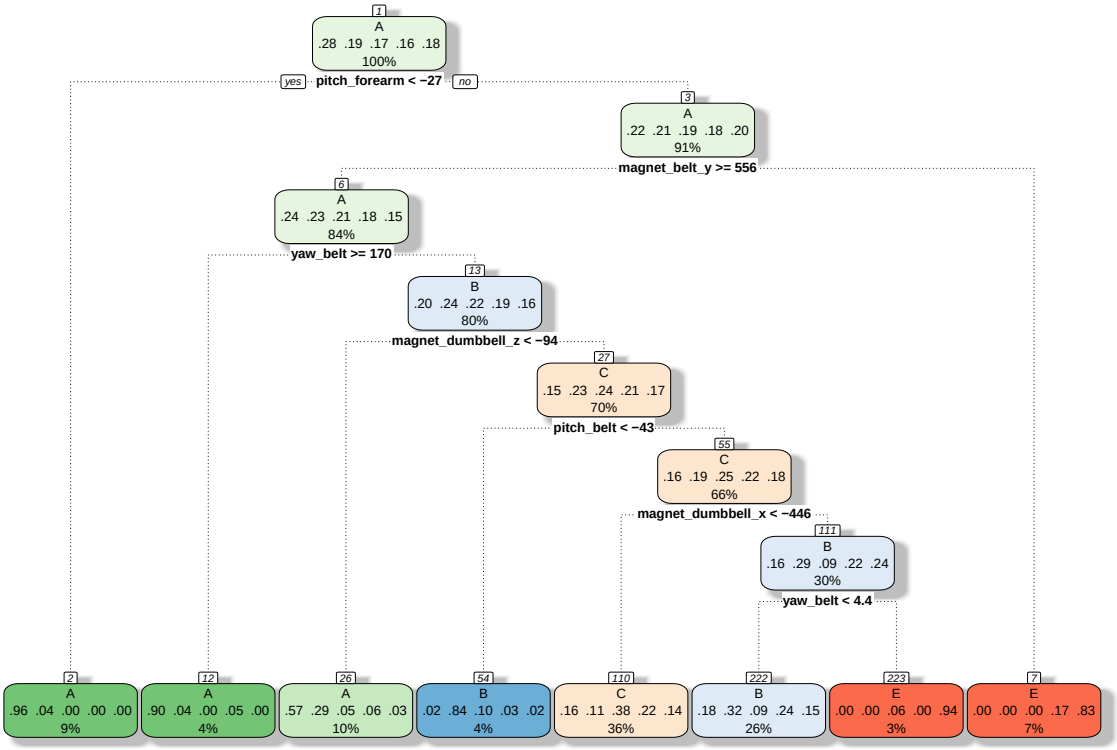
```
## [1] 5885   46
```

rpart prediction

```
set.seed(125)
#rpart train
library(rpart)
rpartFit <- train(classe ~., method='rpart', data=trainDF)
library(rattle)
```

```
## Loading required package: RGtk2
## Rattle: A free graphical interface for data mining with R.
## Version 3.5.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
fancyRpartPlot(rpartFit$finalModel)
```



Rattle 2015−Jul−28 11:23:09 ict

```
#rpart cross validate
rpartPred <- predict(rpartFit, newdata=validateDF)
print(confusionMatrix(rpartPred, validateDF$classe), digits=4)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1047  192   19   47   22
##          B  299  691  184  359  259
##          C  328  251  818  459  273
##          D    0    0    0    0    0
##          E    0    5    5   99  528
##
## Overall Statistics
##
##                Accuracy : 0.524
##                  95% CI : (0.5112, 0.5369)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4005
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.6254   0.6067   0.7973   0.0000  0.48799
```

```
## Specificity            0.9335   0.7680   0.7302   1.0000  0.97731
## Pos Pred Value         0.7890   0.3856   0.3842      NaN  0.82889
## Neg Pred Value         0.8624   0.8905   0.9446   0.8362  0.89444
## Prevalence             0.2845   0.1935   0.1743   0.1638  0.18386
## Detection Rate         0.1779   0.1174   0.1390   0.0000  0.08972
## Detection Prevalence   0.2255   0.3045   0.3618   0.0000  0.10824
## Balanced Accuracy      0.7795   0.6873   0.7637   0.5000  0.73265
```

randomForest prediction

```
#random forest
library('randomForest')
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
#rfFit <- train(classe ~., method='rf', data=trainDF, trControl=trainControl(method="cv",number=5), pro
#rfPred <- predict(rfFit, newdata=validateDF)

#saveRDS(rfFit, file='rfFit.rds')
#saveRDS(rfPred, file='rfPred.rds')

rfFit <- readRDS("/home/ict/rfFit.rds")
rfPred <- readRDS("/home/ict/rfPred.rds")

print(confusionMatrix(rfPred, validateDF$classe), digits=4)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   18    0    0    0
##          B    1 1118    2    1    0
##          C    0    3 1019    9    2
##          D    0    0    5  954    1
##          E    0    0    0    0 1079
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9904, 0.9949)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9816   0.9932   0.9896   0.9972
```

```
## Specificity            0.9957   0.9992   0.9971   0.9988   1.0000
## Pos Pred Value         0.9894   0.9964   0.9864   0.9937   1.0000
## Neg Pred Value         0.9998   0.9956   0.9986   0.9980   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1900   0.1732   0.1621   0.1833
## Detection Prevalence   0.2873   0.1907   0.1755   0.1631   0.1833
## Balanced Accuracy      0.9976   0.9904   0.9951   0.9942   0.9986
```

Conclusion

Predict the testing data

```
#redict testing data set
testPred <- predict(rfFit, newdata=testData)
testPred
```

```
##  [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
```