Machine Learning_Project

Bharat

December 26, 2015

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).

Installing Packages

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.3
```

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

```r
library(ggplot2)
library(lattice)
library(kernlab)
```

```
## Warning: package 'kernlab' was built under R version 3.2.3
```

```r
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.2.3
```

```r
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.2.3
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 4.0.5 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.2.3
```

Downloading datasets

```
traindata <- read.csv('R/pml-training.csv')
validdata <- read.csv('R/pml-testing.csv')
```

Preprocessing Operations

Clearing zero variance

```
set.seed(32768)
nzv <- nearZeroVar(traindata)
traindata <- traindata[-nzv]
validdata <-validdata[-nzv]
```

Clearing irrelevant columns

```
Cl <- grep("name|timestamp|window|X|^max|^min|^std|^amplitude", colnames(traindata), value=F)
traindata <- traindata[,-Cl]
validdata <-validdata[,-Cl]
```

Clearing >95% NAs

```
traindata[traindata==""] <- NA
NAs <- apply(traindata, 2, function(x) sum(is.na(x)))/nrow(traindata)
traindata <- traindata[!(NAs>0.95)]

validdata[validdata==""] <- NA
NAs <- apply(validdata, 2, function(x) sum(is.na(x)))/nrow(validdata)
validdata <- validdata[!(NAs>0.95)]
```

Cross Validation

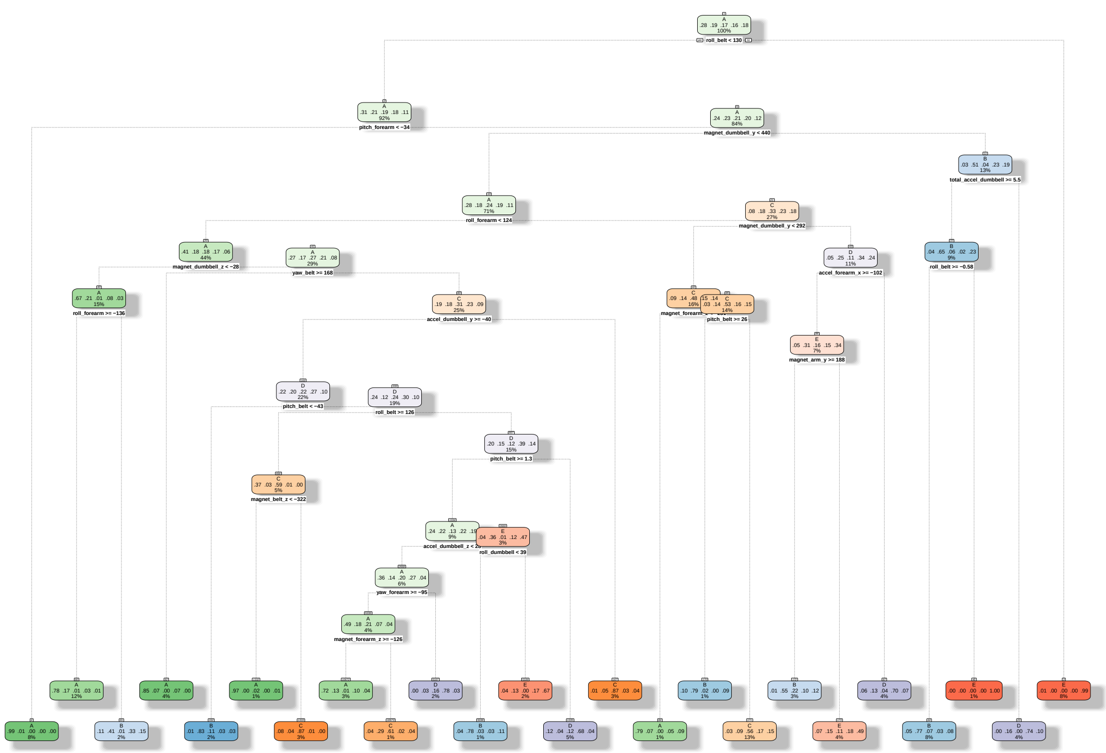
```r
trainidx <- createDataPartition(traindata$classe,p=.9,list=FALSE)
traindata = traindata[trainidx,]
testdata = traindata[-trainidx,]
```

Model Application

CART

```r
model1 <- rpart(classe ~ ., data=traindata, method="class")
fancyRpartPlot(model1)
```

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



Rattle 2015−Dec−26 13:08:04 bharat

```r
#rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)


# Predicting:
prediction1 <- predict(model1, testdata, type = "class")

# Testing:
confusionMatrix(prediction1, testdata$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 463  45   5  22   3
##          B  10 229  33  24  28
##          C   9  31 233  44  48
##          D  16  24  17 190  21
##          E   6  20   8  18 231
##
## Overall Statistics
##
##                Accuracy : 0.757
##                  95% CI : (0.7364, 0.7768)
##     No Information Rate : 0.2835
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6922
##  Mcnemar's Test P-Value : 1.337e-10
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9187   0.6562   0.7872   0.6376   0.6979
## Specificity            0.9411   0.9335   0.9109   0.9473   0.9641
## Pos Pred Value         0.8606   0.7068   0.6384   0.7090   0.8163
## Neg Pred Value         0.9669   0.9175   0.9554   0.9285   0.9331
## Prevalence             0.2835   0.1963   0.1665   0.1676   0.1862
## Detection Rate         0.2604   0.1288   0.1310   0.1069   0.1299
## Detection Prevalence   0.3026   0.1822   0.2053   0.1507   0.1592
## Balanced Accuracy      0.9299   0.7948   0.8490   0.7924   0.8310
```

Random Forest

```r
model2 <- randomForest(classe ~. , data=traindata, method="class")

# Predicting:
prediction2 <- predict(model2, testdata, type = "class")

# Test results on subTesting data set:
confusionMatrix(prediction2, testdata$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 504   0   0   0   0
##          B   0 349   0   0   0
##          C   0   0 296   0   0
##          D   0   0   0 298   0
##          E   0   0   0   0 331
##
```

```
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9979, 1)
##     No Information Rate : 0.2835
##     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             0.2835   0.1963   0.1665   0.1676   0.1862
## Detection Rate         0.2835   0.1963   0.1665   0.1676   0.1862
## Detection Prevalence   0.2835   0.1963   0.1665   0.1676   0.1862
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```
#Predictor importance
importance(model2)
```

```
##                      MeanDecreaseGini
## roll_belt                  1110.50327
## pitch_belt                  638.51628
## yaw_belt                    820.50441
## total_accel_belt            205.12620
## gyros_belt_x                 88.89922
## gyros_belt_y                107.75418
## gyros_belt_z                277.71754
## accel_belt_x                106.01851
## accel_belt_y                110.26759
## accel_belt_z                400.36822
## magnet_belt_x               229.91302
## magnet_belt_y               320.78291
## magnet_belt_z               350.11253
## roll_arm                    285.84205
## pitch_arm                   157.76993
## yaw_arm                     221.17800
## total_accel_arm              89.83605
## gyros_arm_x                 115.12857
## gyros_arm_y                 128.18081
## gyros_arm_z                  51.82840
## accel_arm_x                 206.81301
## accel_arm_y                 135.33070
## accel_arm_z                 122.28596
## magnet_arm_x                250.80318
## magnet_arm_y                197.60666
## magnet_arm_z                164.44469
## roll_dumbbell               363.36475
## pitch_dumbbell              155.10177
```

```
## yaw_dumbbell             224.88719
## total_accel_dumbbell     251.09667
## gyros_dumbbell_x         111.96948
## gyros_dumbbell_y         223.33236
## gyros_dumbbell_z          73.77542
## accel_dumbbell_x         232.33825
## accel_dumbbell_y         353.47274
## accel_dumbbell_z         296.68872
## magnet_dumbbell_x        449.99339
## magnet_dumbbell_y        596.63281
## magnet_dumbbell_z        681.26192
## roll_forearm             563.72357
## pitch_forearm            662.29875
## yaw_forearm              160.38070
## total_accel_forearm       98.29794
## gyros_forearm_x           68.43074
## gyros_forearm_y          115.76433
## gyros_forearm_z           72.76006
## accel_forearm_x          292.74951
## accel_forearm_y          129.26252
## accel_forearm_z          225.34973
## magnet_forearm_x         199.81598
## magnet_forearm_y         201.39052
## magnet_forearm_z         265.29076
```

Predicting on Validation Data Set - FINAL

```
final <- predict(model2, validdata, type="class")
final
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

Write files for submission

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(final)
```