WriteUp Project

WriteUp Project

Practical Machine Learning

Executive Summary

Submission for the predicted answers

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

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).

Data Processing

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```
pmltrain <- read.csv('pml-training.csv')
pmltest <- read.csv('pml-testing.csv')
### Exploratory Data Analysis
### Create training, test and validation sets
library(caret)
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

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

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

```
library(randomForest)
```

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

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

```
trainidx <- createDataPartition(pmltrain$classe,p=.9,list=FALSE)
traindata = pmltrain[trainidx,]
testdata = pmltrain[-trainidx,]
set.seed(32768)
nzv <- nearZeroVar(traindata)
trainnzv <- traindata[-nzv]
testnzv <- testdata[-nzv]
pmltestnzv <- pmltest[-nzv]
```

```
dim(trainnzv)
```

```
## [1] 17662   103
```

```
dim(testnzv)
```

```
## [1] 1960  103
```

```
dim(pmltestnzv)
```

```
## [1]  20 103
```

## [1] 20 102

```
ftridx <- which(lapply(trainnzv,class) %in% c('numeric'))
trainnzv1 <- preProcess(trainnzv[,ftridx], method=c('knnImpute'))
ftridx
```

```
##  [1]   7   8   9  11  13  15  17  18  19  20  21  22  23  24  25  26  27
## [18]  28  29  36  37  38  40  41  42  43  50  52  54  56  57  58  59  60
## [35]  61  62  63  64  66  67  68  69  70  71  72  73  74  75  76  77  78
## [52]  84  85  86  87  88  89  90  91  93  94  95  96 101 102
```

```
trainnzv1
```

```
## Created from 361 samples and 65 variables
##
## Pre-processing:
##   - centered (65)
##   - ignored (0)
##   - 5 nearest neighbor imputation (65)
##   - scaled (65)
```

Call:

preProcess.default(x = trainnzv[, ftridx], method = c("knnImpute"))

Created from 372 samples and 64 variables

Pre-processing: 5 nearest neighbor imputation, scaled, centered

```
pred1 <- predict(trainnzv1, trainnzv[,ftridx])
predtrain <- cbind(trainnzv$classe,pred1)
names(predtrain)[1] <- 'classe'
predtrain[is.na(predtrain)] <- 0

pred2 <- predict(trainnzv1, testnzv[,ftridx])
predtest <- cbind(testnzv$classe, pred2)
names(predtest)[1] <- 'classe'
predtest[is.na(predtest)] <- 0

predpmltest <- predict(trainnzv1,pmltestnzv[,ftridx] )
```

```
dim(predtrain)
```

```
## [1] 17662    66
```

```
dim(predtest)
```

```
## [1] 1960   66
```

```
dim(predpmltest)
```

```
## [1] 20 65
```

Modeling

```
model <- randomForest(classe~.,data=predtrain)
```

```
predtrain1 <- predict(model, predtrain)
print(table(predtrain1, predtrain$classe))
```

```
##
## predtrain1    A    B    C    D    E
##          A 5022    0    0    0    0
##          B    0 3418    0    0    0
##          C    0    0 3080    0    0
##          D    0    0    0 2895    0
##          E    0    0    0    0 3247
```

```
training <- as.data.frame(table(predtrain1, predtrain$classe))
#qplot(training)
```

```
predtest1 <- predict(model, predtest)
print(table(predtest1, predtest$classe))
```

```
##
## predtest1   A   B   C   D   E
##         A 557   3   0   0   1
##         B   0 373   2   0   1
##         C   0   3 338   3   1
##         D   0   0   2 315   1
##         E   1   0   0   3 356
```

```
str(predpmltest)
```

```
## 'data.frame':    20 obs. of  65 variables:
##  $ roll_belt               : num  0.934 -1.009 -1.012 0.966 -1.004 ...
##  $ pitch_belt              : num  1.195 0.204 0.068 -1.875 0.136 ...
##  $ yaw_belt                : num  0.0689 -0.8152 -0.811 1.8207 -0.812 ...
##  $ max_roll_belt           : num  0.0456 -0.8542 -0.8527 1.848 -0.8451 ...
##  $ min_roll_belt           : num  0.0733 -0.826 -0.8242 1.8811 -0.8315 ...
##  $ amplitude_roll_belt     : num  -0.0941 -0.1299 -0.1306 -0.0468 -0.0783 ...
##  $ var_total_accel_belt    : num  -0.332 -0.315 -0.352 -0.371 -0.22 ...
##  $ avg_roll_belt           : num  0.868 -1.06 -1.065 0.905 -1.046 ...
##  $ stddev_roll_belt        : num  -0.439 -0.439 -0.431 -0.363 -0.254 ...
##  $ var_roll_belt           : num  -0.329 -0.331 -0.33 -0.323 -0.26 ...
##  $ avg_pitch_belt          : num  1.105 0.182 0.171 -1.909 0.128 ...
##  $ stddev_pitch_belt       : num  -0.6042 0.1665 -0.0859 -0.4329 0.8165 ...
##  $ var_pitch_belt          : num  -0.387 -0.117 -0.252 -0.32 1.154 ...
##  $ avg_yaw_belt            : num  0.0635 -0.8433 -0.8416 1.8757 -0.8442 ...
##  $ stddev_yaw_belt         : num  -0.0914 -0.1139 -0.1157 -0.0643 -0.0715 ...
##  $ var_yaw_belt            : num  -0.0685 -0.0686 -0.0686 -0.0682 -0.0678 ...
##  $ gyros_belt_x            : num  -2.386 -0.261 0.27 0.56 0.174 ...
##  $ gyros_belt_y            : num  -0.764 -0.764 -0.251 0.904 -0.251 ...
##  $ gyros_belt_z            : num  -1.369 0.253 0.669 -0.121 0.544 ...
##  $ roll_arm                : num  0.311 -0.249 -0.249 -1.75 0.798 ...
##  $ pitch_arm               : num  -0.756 0.149 0.149 1.939 0.238 ...
##  $ yaw_arm                 : num  2.51 0.0104 0.0104 -1.9837 1.4427 ...
##  $ var_accel_arm           : num  0.3506 -0.4836 0.0324 -0.4409 -0.7112 ...
##  $ gyros_arm_x             : num  -0.846 -0.606 1.029 0.089 -1.002 ...
##  $ gyros_arm_y             : num  0.863 1.296 -1.289 -0.295 1.225 ...
##  $ gyros_arm_z             : num  -0.81 -1.26 1.55 1.17 -1.46 ...
##  $ max_picth_arm           : num  1.223 -0.499 -0.499 -1.245 0.472 ...
##  $ min_pitch_arm           : num  1.279 0.551 0.551 -1.303 0.946 ...
##  $ amplitude_pitch_arm     : num  0.103 -1.041 -1.041 -0.103 -0.382 ...
##  $ roll_dumbbell           : num  -0.596 0.439 0.476 0.276 -1.794 ...
##  $ pitch_dumbbell          : num  0.963 -1.162 -1.099 -0.523 -1.155 ...
##  $ yaw_dumbbell            : num  1.505 -0.936 -0.932 -1.273 -0.194 ...
##  $ max_roll_dumbbell       : num  0.699 -1.084 -1.029 -0.816 -0.405 ...
##  $ max_picth_dumbbell      : num  1.1198 -1.0778 -0.9764 -1.3188 0.0683 ...
##  $ min_roll_dumbbell       : num  1.4485 -0.498 -0.632 0.0989 -0.2859 ...
##  $ min_pitch_dumbbell      : num  1.788 -0.724 -0.717 -1.013 0.18 ...
##  $ amplitude_roll_dumbbell : num  -0.289 -0.652 -0.519 -0.79 -0.18 ...
##  $ amplitude_pitch_dumbbell: num  -0.434 -0.72 -0.582 -0.736 -0.108 ...
##  $ var_accel_dumbbell      : num  -0.228 -0.208 -0.184 -0.275 -0.27 ...
##  $ avg_roll_dumbbell       : num  -0.421 0.59 0.642 0.393 -0.739 ...
##  $ stddev_roll_dumbbell    : num  -0.343 -0.58 -0.503 -0.641 -0.14 ...
##  $ var_roll_dumbbell       : num  -0.383 -0.42 -0.409 -0.44 -0.263 ...
##  $ avg_pitch_dumbbell      : num  1.282 -1.082 -1.078 -0.569 -0.578 ...
##  $ stddev_pitch_dumbbell   : num  -0.3537 -0.6002 -0.5851 -0.7959 0.0348 ...
##  $ var_pitch_dumbbell      : num  -0.4085 -0.4464 -0.4684 -0.5041 -0.0581 ...
##  $ avg_yaw_dumbbell        : num  1.5407 -0.9727 -0.939 -1.2741 0.0906 ...
##  $ stddev_yaw_dumbbell     : num  -0.463 -0.68 -0.585 -0.715 -0.108 ...
##  $ var_yaw_dumbbell        : num  -0.41 -0.444 -0.424 -0.453 -0.222 ...
##  $ gyros_dumbbell_x        : num  0.3021 0.1128 0.1444 -0.0386 0.0813 ...
##  $ gyros_dumbbell_y        : num  0.024 0.00792 0.15268 -0.10468 -0.82852 ...
##  $ gyros_dumbbell_z        : num  -0.2007 -0.2422 -0.0885 0.0735 -0.1384 ...
##  $ magnet_dumbbell_z       : num  -0.7314 -0.5887 -0.0393 0.0463 1.8941 ...
##  $ roll_forearm            : num  0.99 0.694 0.897 -0.315 -1.945 ...
##  $ pitch_forearm           : num  1.371 -1.01 -1.544 -0.384 -0.461 ...
##  $ yaw_forearm             : num  1.325 0.84 0.714 -0.188 -0.652 ...
##  $ max_picth_forearm       : num  0.6756 0.5541 0.4478 -0.8525 -0.0794 ...
##  $ min_pitch_forearm       : num  0.843 0.73 0.81 0.531 -0.742 ...
##  $ amplitude_roll_forearm  : num  1.1429 0.3779 0.1853 -0.9529 -0.0317 ...
##  $ amplitude_pitch_forearm : num  -0.192 -0.187 -0.316 -0.947 0.503 ...
##  $ var_accel_forearm       : num  -0.635 -0.372 -0.772 -0.738 0.742 ...
##  $ gyros_forearm_x         : num  0.898 1.481 0.038 1.881 -1.39 ...
##  $ gyros_forearm_y         : num  -1.076 -0.9 -0.276 0.188 0.944 ...
##  $ gyros_forearm_z         : num  -0.4057 -0.1826 0.0677 0.8948 0.3507 ...
##  $ magnet_forearm_y        : num  0.0759 0.8064 0.6238 0.7907 -2.2926 ...
##  $ magnet_forearm_z        : num  0.604 1.296 1.053 0.345 -0.817 ...
```

```
predanswers <- predict(model, predpmltest)
predanswers
```

```
##  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  C  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

Results

```
predanswers <- predict(model, predpmltest)
predanswers
```

```
##  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  C  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

```
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(as.character(predanswers))
```