Prediction Assignment Writeup

VM

March 5, 2016

OverView

The goal of your project is to predict the manner in which exercise is done.
This is the "classe" variable in the training set. You may use any of the other variables to predict with.

1)You should create a report describing how you built your model
2)how you used cross validation
3)what you think the expected out of sample error
4)why you made the choices you did

DATA SAMPLE INFO

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

****Loading requried libraries****

```
library(Hmisc)
library(caret)
library(randomForest)
library(gbm)
set.seed(2016)
```

Data Loading and cleansing

```
trainDataRaw<- read.csv("pml-training.csv", na.strings = c("","NA", "NULL"))
testDataRaw<- read.csv("pml-testing.csv", na.strings = c("","NA", "NULL"))
```

```
naprops <- colSums(is.na(trainDataRaw))/nrow(trainDataRaw)
```

Exploration showed they are lot of NA's, Null

```
napercent <- colSums(is.na(trainDataRaw))/nrow(trainDataRaw)
head(napercent)
```

Let see the amount NAs and Null present

```
##                    X            user_name raw_timestamp_part_1
##                    0                    0                    0
## raw_timestamp_part_2       cvtd_timestamp           new_window
##                    0                    0                    0
```

there are about 98% of NA's, and we exclude this to make prediction data clean, excluding NNA

```
ptrainDataNNA <- trainDataRaw[,colSums(is.na(trainDataRaw)) == 0]
```

we exclude these columns, these colums which are not used for prediction analysis. >excluding data with are not use full for analysis NAs and etc
$ X : int 1 2 3 4 5 6 7 8 9 10 . . .
$ user_name : Factor w/ 6 levels "adelmo","carlitos",..: 2 2 2 2 2 2 2 2 2 2 . . .
$ raw_timestamp_part_1 : int 1323084231 1323084231 1323084231 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232
$ raw_timestamp_part_2 : int 788290 808298 820366 120339 196328 304277 368296 440390 484323 484434 . . .
$ cvtd_timestamp : Factor w/ 20 levels "02/12/2011 13:32",..: 9 9 9 9 9 9 9 9 9 9 . . .
$ new_window : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 . . .
$ num_window : int 11 11 11 12 12 12 12 12 12 12 . . .

```
predicitiontrain <- ptrainDataNNA[,8:length(ptrainDataNNA[1,])]
```

```
nzvColumn <-  which(nearZeroVar(predicitiontrain, saveMetrics = TRUE)$nzv == FALSE)
predicitiontrainNZV <- predicitiontrain[,nzvColumn]
```

check for Non zero variance predictors, and exclude them since not used for prediction

exclude highly correlated variables helps us to build model with required varaibles, 90% in this test case.

```
corrMatrix<- cor(predicitiontrainNZV[,sapply(predicitiontrainNZV, is.numeric)])
highcorrvb<- findCorrelation(corrMatrix, cutoff = 0.9, verbose = TRUE)
```

Get the correlation between each varaible and get high correlations and remove them

```
## 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
```

```r
predicitionDataSet<- predicitiontrainNZV[,-highcorrvb]
dim(predicitionDataSet)
```

```
## [1] 19622    46
```

Model for prediction

```r
inTrain <- createDataPartition(predicitionDataSet$classe, p = 3/4, list = FALSE)
training <- predicitionDataSet[inTrain,]
testing <- predicitionDataSet[-inTrain,]
```

```r
## analyze the data with caret package
modrpart <- train(classe ~., method = "rpart", data = training)
```

split the data for cross validation

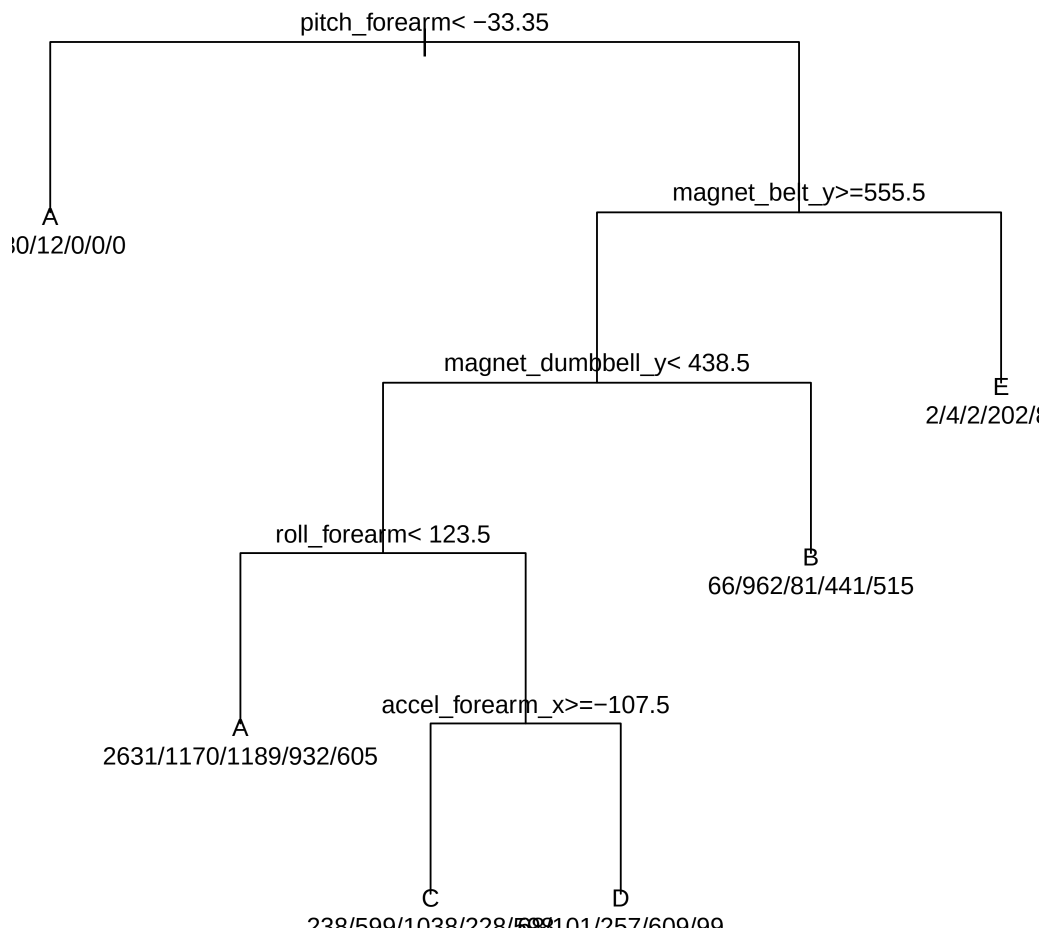
```
## Loading required package: rpart
```

```r
print(modrpart$finalModel)
```

```
## n= 14718
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 14718 10533 A (0.28 0.19 0.17 0.16 0.18)
##    2) pitch_forearm< -33.35 1192    12 A (0.99 0.01 0 0 0) *
##    3) pitch_forearm>=-33.35 13526 10521 A (0.22 0.21 0.19 0.18 0.2)
##      6) magnet_belt_y>=555.5 12427  9424 A (0.24 0.23 0.21 0.18 0.15)
##       12) magnet_dumbbell_y< 438.5 10362  7425 A (0.28 0.18 0.24 0.17 0.13)
##         24) roll_forearm< 123.5 6527  3896 A (0.4 0.18 0.18 0.14 0.093) *
##         25) roll_forearm>=123.5 3835  2540 C (0.08 0.18 0.34 0.22 0.18)
##           50) accel_forearm_x>=-107.5 2701  1663 C (0.088 0.22 0.38 0.084 0.22) *
##           51) accel_forearm_x< -107.5 1134   525 D (0.06 0.089 0.23 0.54 0.087) *
##       13) magnet_dumbbell_y>=438.5 2065  1103 B (0.032 0.47 0.039 0.21 0.25) *
##      7) magnet_belt_y< 555.5 1099   210 E (0.0018 0.0036 0.0018 0.18 0.81) *
```

Plot Classification Tree

Classification Tree




```r
predrpart <- predict(modrpart, testing)
table(predrpart, testing$classe)
```

check accuracy, which is near 50% not encouraging to prediction model

```
##
## predrpart    A    B    C    D    E
##         A 1281  400  395  334  195
##         B   21  331   28  131  164
##         C   74  187  340   84  204
```

```
##         D   18   31   92  191   37
##         E    1    0    0   64  301
```

```r
confusionMatrix(testing$classe, predrpart)$overall['Accuracy']
```

```
##  Accuracy
## 0.4983687
```

GDM (Generalized boosted Regression Model) Prediction

The predict returns back the probability for each classe, Below for each row we pick the one with largest probability,

```r
modgbm <- gbm(classe ~., data = training, distribution = "multinomial", n.trees = 200, interaction.dep
predgbm <- predict(modgbm, n.trees = 200, newdata= testing, type = 'response')
maxpredgbm <- apply(predgbm, 1, which.max)
```

```r
## Since 1~5 means A ~ E, we rename them below
maxpredgbm[which(maxpredgbm == 1)] <- "A"
maxpredgbm[which(maxpredgbm == 2)] <- "B"
maxpredgbm[which(maxpredgbm == 3)] <- "C"
maxpredgbm[which(maxpredgbm == 4)] <- "D"
maxpredgbm[which(maxpredgbm == 5)] <- "E"
maxpredgbm <- as.factor(maxpredgbm)
```

```r
# check the accuracy using confusionMatrix
confusionMatrix(testing$classe, maxpredgbm)$overall['Accuracy']
```

the accuracy is about 77%.

```
##  Accuracy
## 0.7832382
```

Random Forest Prediction, the accurancy obout 99%

```r
library(randomForest)
modrf <- randomForest(classe~., data = training, ntree=100, importance=TRUE, prox = TRUE)
predrf <- predict(modrf, testing)
table(predrf, testing$classe)
```

```
##
## predrf    A    B    C    D    E
##      A 1393    2    0    0    0
```

```
##          B    2  941    1    0    1
##          C    0    6  854    4    0
##          D    0    0    0  800    0
##          E    0    0    0    0  900
```

```r
confusionMatrix(testing$classe, predrf)$overall['Accuracy']
```

```
##  Accuracy
## 0.9967374
```