Machine learning project"

Introduction

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, 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

Downloading data for hte project

```
download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",
              "train_set.csv",mode="wb")
download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
              "test_set.csv", mode="wb")

train_set <- read.csv("train_set.csv",na.strings=c("NA","#DIV/0!", ""), row.names = 1)
test_set <- read.csv("test_set.csv",na.strings=c("NA","#DIV/0!", ""), row.names = 1)

dim(train_set)
```

```
## [1] 19622   159
```

```
dim(test_set)
```

```
## [1]  20 159
```

```
table(train_set$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Data Processing

First round of data processing will include cleaning data.

```
##check for NAs
nasPerColumn <- apply(train_set, 2, function(x) length(which(is.na(x))))
nasPerColumn
```

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

```
##                roll_belt              pitch_belt                yaw_belt
##                        0                       0                       0
##         total_accel_belt      kurtosis_roll_belt     kurtosis_picth_belt
##                        0                   19226                   19248
##        kurtosis_yaw_belt      skewness_roll_belt    skewness_roll_belt.1
##                    19622                   19225                   19248
##        skewness_yaw_belt           max_roll_belt          max_picth_belt
##                    19622                   19216                   19216
##             max_yaw_belt           min_roll_belt          min_pitch_belt
##                    19226                   19216                   19216
##             min_yaw_belt     amplitude_roll_belt    amplitude_pitch_belt
##                    19226                   19216                   19216
##       amplitude_yaw_belt    var_total_accel_belt           avg_roll_belt
##                    19226                   19216                   19216
##         stddev_roll_belt           var_roll_belt          avg_pitch_belt
##                    19216                   19216                   19216
##        stddev_pitch_belt          var_pitch_belt            avg_yaw_belt
##                    19216                   19216                   19216
##          stddev_yaw_belt            var_yaw_belt            gyros_belt_x
##                    19216                   19216                       0
##             gyros_belt_y            gyros_belt_z            accel_belt_x
##                        0                       0                       0
##             accel_belt_y            accel_belt_z           magnet_belt_x
##                        0                       0                       0
##            magnet_belt_y           magnet_belt_z                roll_arm
##                        0                       0                       0
##                pitch_arm                 yaw_arm         total_accel_arm
##                        0                       0                       0
##            var_accel_arm            avg_roll_arm         stddev_roll_arm
##                    19216                   19216                   19216
##             var_roll_arm           avg_pitch_arm        stddev_pitch_arm
##                    19216                   19216                   19216
##            var_pitch_arm             avg_yaw_arm          stddev_yaw_arm
##                    19216                   19216                   19216
##              var_yaw_arm             gyros_arm_x             gyros_arm_y
##                    19216                       0                       0
##              gyros_arm_z             accel_arm_x             accel_arm_y
##                        0                       0                       0
##              accel_arm_z            magnet_arm_x            magnet_arm_y
##                        0                       0                       0
##             magnet_arm_z       kurtosis_roll_arm      kurtosis_picth_arm
##                        0                   19294                   19296
##         kurtosis_yaw_arm       skewness_roll_arm      skewness_pitch_arm
##                    19227                   19293                   19296
##         skewness_yaw_arm            max_roll_arm           max_picth_arm
##                    19227                   19216                   19216
##              max_yaw_arm            min_roll_arm           min_pitch_arm
##                    19216                   19216                   19216
##              min_yaw_arm      amplitude_roll_arm     amplitude_pitch_arm
##                    19216                   19216                   19216
##        amplitude_yaw_arm           roll_dumbbell          pitch_dumbbell
##                    19216                       0                       0
##             yaw_dumbbell  kurtosis_roll_dumbbell kurtosis_picth_dumbbell
##                        0                   19221                   19218
```

```
##    kurtosis_yaw_dumbbell   skewness_roll_dumbbell  skewness_pitch_dumbbell
##                    19622                    19220                    19217
##    skewness_yaw_dumbbell        max_roll_dumbbell        max_picth_dumbbell
##                    19622                    19216                    19216
##         max_yaw_dumbbell        min_roll_dumbbell        min_pitch_dumbbell
##                    19221                    19216                    19216
##         min_yaw_dumbbell  amplitude_roll_dumbbell amplitude_pitch_dumbbell
##                    19221                    19216                    19216
##   amplitude_yaw_dumbbell      total_accel_dumbbell        var_accel_dumbbell
##                    19221                        0                    19216
##        avg_roll_dumbbell      stddev_roll_dumbbell         var_roll_dumbbell
##                    19216                    19216                    19216
##       avg_pitch_dumbbell     stddev_pitch_dumbbell        var_pitch_dumbbell
##                    19216                    19216                    19216
##         avg_yaw_dumbbell       stddev_yaw_dumbbell          var_yaw_dumbbell
##                    19216                    19216                    19216
##         gyros_dumbbell_x         gyros_dumbbell_y          gyros_dumbbell_z
##                        0                        0                        0
##         accel_dumbbell_x         accel_dumbbell_y          accel_dumbbell_z
##                        0                        0                        0
##        magnet_dumbbell_x        magnet_dumbbell_y         magnet_dumbbell_z
##                        0                        0                        0
##             roll_forearm            pitch_forearm              yaw_forearm
##                        0                        0                        0
##    kurtosis_roll_forearm   kurtosis_picth_forearm     kurtosis_yaw_forearm
##                    19300                    19301                    19622
##    skewness_roll_forearm   skewness_pitch_forearm     skewness_yaw_forearm
##                    19299                    19301                    19622
##         max_roll_forearm         max_picth_forearm          max_yaw_forearm
##                    19216                    19216                    19300
##         min_roll_forearm         min_pitch_forearm          min_yaw_forearm
##                    19216                    19216                    19300
##   amplitude_roll_forearm  amplitude_pitch_forearm    amplitude_yaw_forearm
##                    19216                    19216                    19300
##      total_accel_forearm         var_accel_forearm         avg_roll_forearm
##                        0                    19216                    19216
##      stddev_roll_forearm          var_roll_forearm        avg_pitch_forearm
##                    19216                    19216                    19216
##     stddev_pitch_forearm         var_pitch_forearm          avg_yaw_forearm
##                    19216                    19216                    19216
##       stddev_yaw_forearm           var_yaw_forearm          gyros_forearm_x
##                    19216                    19216                        0
##          gyros_forearm_y           gyros_forearm_z          accel_forearm_x
##                        0                        0                        0
##          accel_forearm_y           accel_forearm_z         magnet_forearm_x
##                        0                        0                        0
##         magnet_forearm_y          magnet_forearm_z                   classe
##                        0                        0                        0
```

```
##remove columns that contain mostly NA values (90%)
train_set <- train_set[,which(nasPerColumn <  nrow(train_set)*0.9)]

library(caret)
```

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

```
##remove near zero values
nearZeroColumns <- nearZeroVar(train_set, saveMetrics = TRUE)
train_set <- train_set[, nearZeroColumns$nzv==FALSE]

##subset the taining set to only predictor columns
train_set   <-train_set[,-c(1:7)]
test_set <-test_set[,-c(1:7)]

##Classe as factor
train_set$classe <- factor(train_set$classe)
```

Now let's do preprocessing for the model training

```
library(caret)
set.seed(23232)
inTrain <- createDataPartition(train_set$classe, p=0.75, list = FALSE)
train <- train_set[inTrain,]
test <- train_set[-inTrain,]
```

Model Training and validation

First, we will build descision tree model.

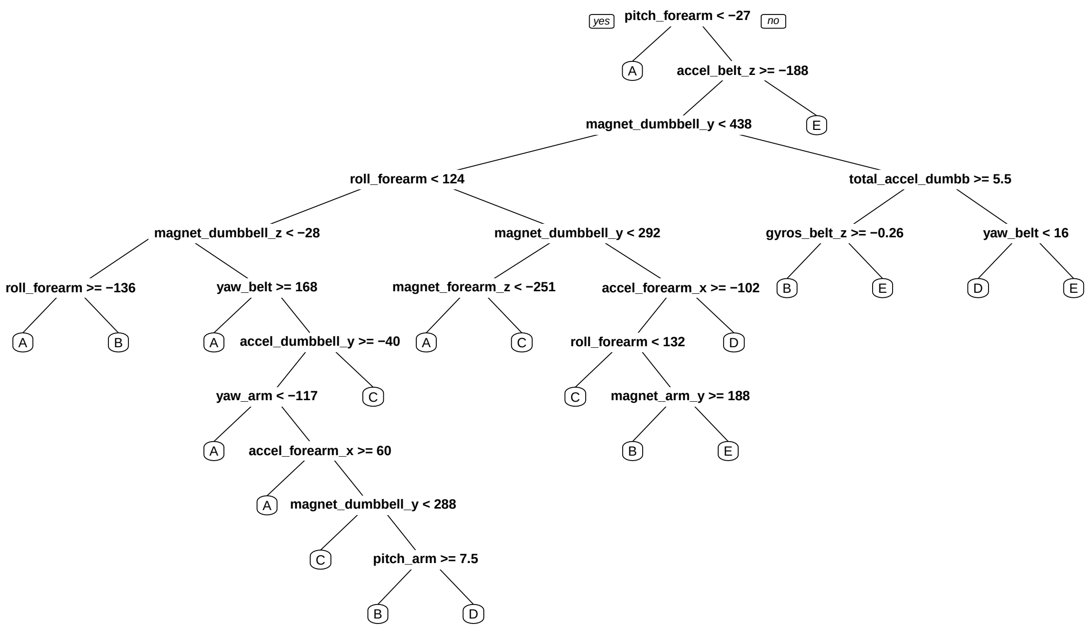
```
library(randomForest)
```

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

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

firstmodel <- rpart(classe ~ ., data=train, method="class")
prediction <- predict(firstmodel, test, type = "class")
rpart.plot(firstmodel, main="Classification Tree")
```

Classification Tree




```r
##review confusin matrix
confusionMatrix(prediction, test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1256  179   36   62   64
##          B   37  423   66   88  133
##          C   38  211  732  209  219
##          D   49  104   14  407   61
##          E   15   32    7   38  424
##
## Overall Statistics
##
##                Accuracy : 0.6611
##                  95% CI : (0.6477, 0.6743)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5695
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
```

```
## Sensitivity            0.9004  0.44573   0.8561  0.50622  0.47059
## Specificity            0.9028  0.91808   0.8328  0.94439  0.97702
## Pos Pred Value         0.7865  0.56627   0.5195  0.64094  0.82171
## Neg Pred Value         0.9580  0.87347   0.9648  0.90700  0.89129
## Prevalence             0.2845  0.19352   0.1743  0.16395  0.18373
## Detection Rate         0.2561  0.08626   0.1493  0.08299  0.08646
## Detection Prevalence   0.3257  0.15232   0.2873  0.12949  0.10522
## Balanced Accuracy      0.9016  0.68191   0.8445  0.72530  0.72380
```

Now we will build random forest model.

```r
scondmodel <- randomForest(classe ~ ., data=train, method="class")
prediction <- predict(scondmodel, test, type = "class")

##review confusin matrix
confusionMatrix(prediction, test$classe)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction    A    B    C    D    E
##          A 1392    7    0    0    0
##          B    2  939    5    0    0
##          C    0    3  849   18    0
##          D    0    0    1  786    3
##          E    1    0    0    0  898
##
## Overall Statistics
##
##                Accuracy : 0.9918
##                  95% CI : (0.9889, 0.9942)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9897
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9978   0.9895   0.9930   0.9776   0.9967
## Specificity            0.9980   0.9982   0.9948   0.9990   0.9998
## Pos Pred Value         0.9950   0.9926   0.9759   0.9949   0.9989
## Neg Pred Value         0.9991   0.9975   0.9985   0.9956   0.9993
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2838   0.1915   0.1731   0.1603   0.1831
## Detection Prevalence   0.2853   0.1929   0.1774   0.1611   0.1833
## Balanced Accuracy      0.9979   0.9938   0.9939   0.9883   0.9982
```

We see that reandom forest model accuracy is better then the decision tree model accuracy, so we should be using that model to predict results for the test_set file.

Test Set predictions

```
test_predictions <- predict(scondmodel, test_set, type="class")

##the list of answers
test_predictions
```

```
##  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
```

Creating submission files.

```
# 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(test_predictions)
```