Introduction

Introduction

It is now relatively in easy and expensive to collect data regarding ones activity and performance with the use of Jawbone Up, Nike FuelBand, and Fitbit monitoring devices. These type of devices are part of Quantified Self - movement, whereby a group of enthusiasts take measurements of themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. In this project, we will use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise. The goal is to look at how well well quantified activities are performed.

Data Preprocessing

```
pkgTest <- function(x)
{
  if (!require(x,character.only = TRUE))
  {
    install.packages(x,dep=TRUE)
    if(!require(x,character.only = TRUE)) stop("Package not found")
  }
}

pkgTest("caret");pkgTest("randomForest");pkgTest("corrplot");pkgTest("rpart");
pkgTest("ggplot2");pkgTest("gridExtra");pkgTest("reshape");pkgTest("gplots");
pkgTest("corrplot");pkgTest("rpart.plot")
```

```
## Loading required package: rpart.plot
```

Download and Read Data

```
training.url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test.url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
path <- getwd()
download.file(training.url, destfile=paste(path,"pml_Train",sep="/"), method="curl")
download.file(test.url, destfile=paste(path,"pml_Test",sep="/"), method="curl")
training <- read.csv("pml_Train");test <- read.csv("pml_Test")
dim(training);dim(test)
```

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

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

Our goal is to predict the "classe" variable. The training and test sets contain the same number of variables (160). However, the training set contains 19622 observations while the test set contains 20 observations.

Clean Data

We will clean the data: delete missing values and delete useless variables.

```
classe.training<-training$classe
clean.train <- training[,c(-1,-3,-4,-5,-6,-7)] #timestamp variables and and window are not necessary
clean.test <- test[,c(-1,-3,-4,-5,-6,-7)]
clean.train <- clean.train[, colSums(is.na(clean.train)) == 0];dim(clean.train) #Deletes columns with missing (NA)
```

```
## [1] 19622    87
```

```
clean.test <- clean.test[, colSums(is.na(clean.test)) == 0];dim(clean.test)
```

```
## [1] 20 54
```

```
clean.train<-clean.train[,sapply(clean.train,is.numeric)];dim(clean.train)
```

```
## [1] 19622    52
```

```
clean.test<-clean.test[,sapply(clean.test,is.numeric)];dim(clean.test)
```

```
## [1] 20 53
```

```
clean.train <- clean.train[,colnames(clean.train) %in% colnames(clean.test)];dim(clean.train)
```

```
## [1] 19622    52
```

```
clean.train$classe<-classe.training;dim(clean.train)
```

```
## [1] 19622    53
```

The clean training and test sets now contain 53 variables instead of 160.

Split Data

We will split the clean training set into 70:30 training to validation sets. The validation set will be utilized to perform cross validation.

```
set.seed(3433) # For reproducibile purpose
inTrain <- createDataPartition(clean.train$classe, p=3/4, list=FALSE)
train <- clean.train[inTrain, ];testing <- clean.train[-inTrain, ]
```

Model Data

We fit a predictive model for activity recognition using Random Forest and use 5-fold cross validation.

```
trainRF <- trainControl(method="cv", 5)
model.trainRF <- train(classe ~ ., data=train, method="rf", trControl=trainRF, ntree=3)
model.trainRF
```

```
## Random Forest
##
## 14718 samples
##    59 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 11775, 11774, 11775, 11775, 11773
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD   Kappa SD
##    2    0.9203697  0.8992664  0.0207076530  0.0262005770
##   41    0.9998642  0.9998282  0.0001860155  0.0002352780
##   81    0.9995923  0.9994843  0.0004430355  0.0005603675
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 41.
```

```
predictRF <- predict(model.trainRF, testing);confusionMatrix(testing$classe, predictRF)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    1  948    0    0    0
##          C    0    0  855    0    0
##          D    0    0    0  804    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9998
##                  95% CI : (0.9989, 1)
##     No Information Rate : 0.2847
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9997
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   0.9997   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   0.9989   1.0000   1.0000   1.0000
## Neg Pred Value         0.9997   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2847   0.1933   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1933   0.1743   0.1639   0.1837
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9996   0.9999   1.0000   1.0000   1.0000
```

```
accuracy <- postResample(predictRF, testing$classe);accuracy
```

```
##  Accuracy     Kappa
## 0.9997961 0.9997421
```

```
sample.error <- 1 - as.numeric(confusionMatrix(testing$classe, predictRF)$overall[1]);sample.error
```

```
## [1] 0.0002039152
```

```
# out of sample error can reasonably be expected to be equal to 1 minus the accuracy.
```

The estimated model accuracy is 96.39% and the estimated out-of-sample error is 3.6%.

Predict for Original Test Data

Now, we apply the model to the original test set and remove `problem_id` column.

```
predictTest <- predict(model.trainRF, clean.test[, -length(names(clean.test))]);
```

```
## Error in eval(expr, envir, enclos): object 'X' not found
```

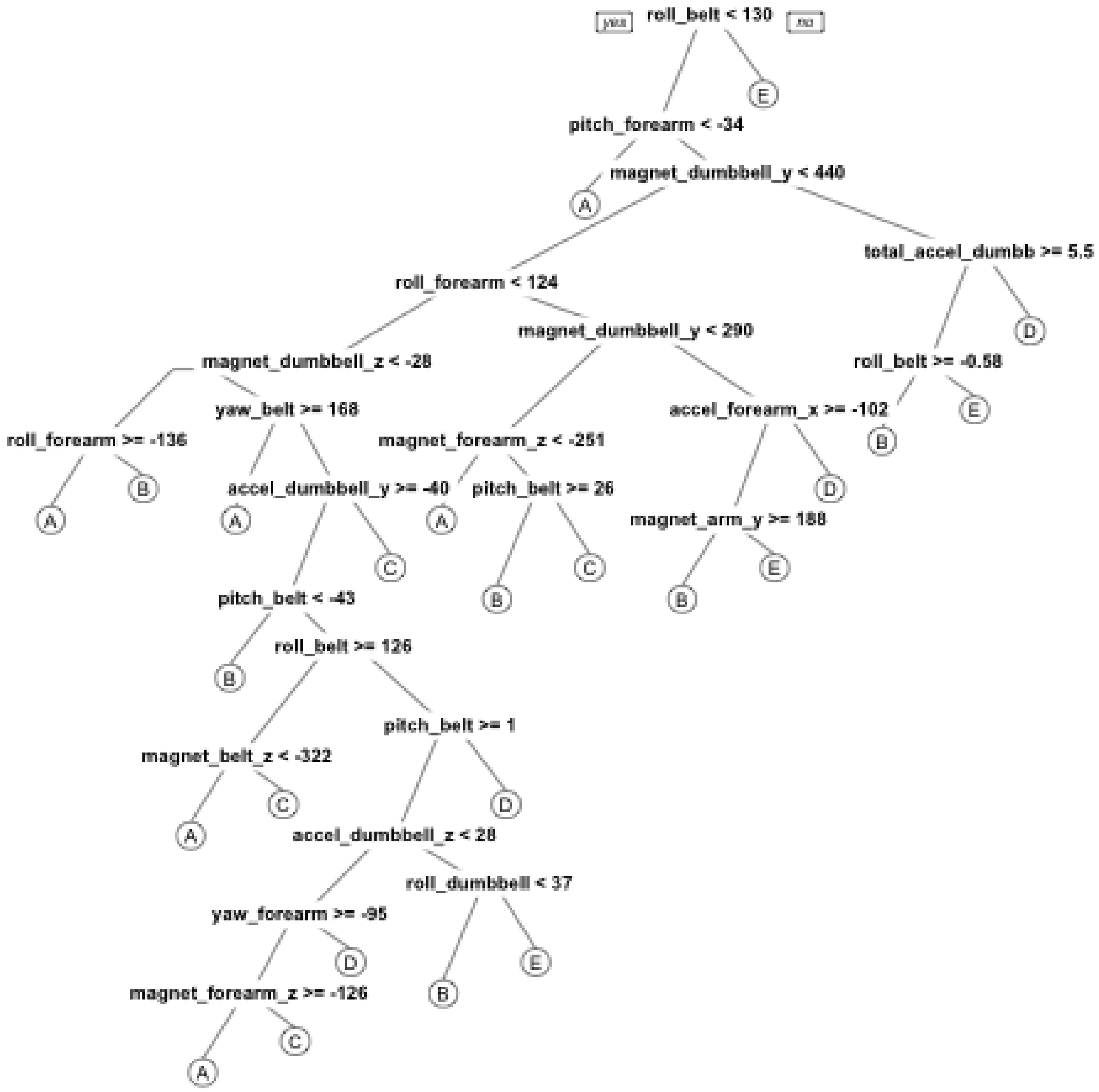
```
predictTest
```

```
## Error in eval(expr, envir, enclos): object 'predictTest' not found
```

Appendix: Figures

Decision Tree

```
Tree <- rpart(classe ~ ., data=clean.train, method="class")
prp(Tree) # Tree model
```


yes
roll_belt < 130
no
E
pitch_forearm < -34
magnet_dumbbell_y < 440
A
total_accel_dumbb >= 5.5
roll_forearm < 124
magnet_dumbbell_y < 290
D
magnet_dumbbell_z < -28
roll_belt >= -0.58
yaw_belt >= 168
accel_forearm_x >= -102
E
B
roll_forearm >= -136
magnet_forearm_z < -251
B
accel_dumbbell_y >= -40
pitch_belt >= 26
D
A
A
A
magnet_arm_y >= 188
C
C
E
pitch_belt < -43
B
B
roll_belt >= 126
B
pitch_belt >= 1
magnet_belt_z < -322
C
D
A
accel_dumbbell_z < 28
roll_dumbbell < 37
yaw_forearm >= -95
D
E
magnet_forearm_z >= -126
B
C
A