Practical Machine Learning Assignment

Apurv Priyam

Feb 27, 2017

Synopsis

Devices such as Jawbone Up, Nike FuelBand, and Fitbit can now help collect a large amount of data about personal activity relatively inexpensively. Often this data is used to measure what kind of activity has been carried out.

This particular exercise works to quantify how well a particular activity has been carried out. With one class depicting right way to do that activity and other five classes representing the most commonly carried out mistakes. We will data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and or incorrectly in 4 different ways(5 classes in total). 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).

Understanding Data

The data set is really large in terms of the features available and are all numeric in nature given then that they are either accelerometer data or their derivatives.

We also have some timestamps data and username and the last column represents the class variable. We convert all data from accelerometers and non-time stamps/class variable to numeric form.

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.2

pml_training <- read.csv("~/Coursera/practical_machine_learning/pml-
training.csv")
pml_testing <- read.csv("~/Coursera/practical_machine_learning/pml-
testing.csv")
for(i in 7:159)
{
  pml_training[,i] <- as.numeric(pml_training[,i])
}
```

Now that we have converted all data points to numeric, we have NA values for certain columns which we initally of character type and empty with no data points.

Pre-Processing

Given the number of features, the first step would be to clean the data. Not using PCA, because want to retain the nature of the original columns.

```
Total_var <- 0
for(i in 8:159)
{
  Int_var <- var(pml_training[,i],na.rm = TRUE)
  if(!is.na(Int_var))
  {
    Total_var <- Total_var + Int_var
  }
}
```

So as to weed out colmuns of data with out suficient variance in them, we compute the contribution of each column to variance. We simply sum up the variance in each column to get the Total Variance available(Assumed, that the columns are independent).

Now, compute the contribution of each variable to the Total Variance. If this variance is less 1%, we straight away reject these columns.

```
col_name <- colnames(pml_training)[8:159]
var_col <- numeric(length(col_name))
for(i in 8:159)
{
  var_col[(i-7)] <- (var(pml_training[,i],na.rm = TRUE)/(Total_var))*100
}

Var_Data <- data.frame(name=col_name,variance_percent=var_col)
```

We eliminate any variable with less than 1% of variance contribution to the total variance pool. The total number of variables with contribution greater than 1%. We also remove any feature with NA value.

```
## [1] 25
```

Thereby the relevant columns are :

```
##  [1] var_yaw_belt            var_roll_arm
##  [3] var_pitch_arm           var_yaw_arm
##  [5] accel_arm_x             accel_arm_z
##  [7] magnet_arm_x            magnet_arm_y
##  [9] magnet_arm_z            var_roll_dumbbell
## [11] var_pitch_dumbbell      var_yaw_dumbbell
## [13] magnet_dumbbell_x       magnet_dumbbell_y
## [15] magnet_dumbbell_z       amplitude_pitch_forearm
## [17] var_roll_forearm        var_pitch_forearm
## [19] var_yaw_forearm         accel_forearm_x
## [21] accel_forearm_y         accel_forearm_z
## [23] magnet_forearm_x        magnet_forearm_y
```

```
## [25] magnet_forearm_z
## 152 Levels: accel_arm_x accel_arm_y accel_arm_z ... yaw_forearm
```

Now that we have weeded out low contributors. We can see if have high correlation ones among the 25 features left.

Modelling

Found out that several of the 25 variables are actually variance themselves, also many of them have plenty of missing values because they might be variance for a certain frame of data over a period of time. Since, I plan to model using Random Forest (Bagging Tree), no need for feature selection.

All variance columns are also removed from the 25, leaving us with 12 features in total. An RF model was built using them and tested on the testing data set.

The performance was decent since the 20 questions based on the testing data set, all came out to be right.

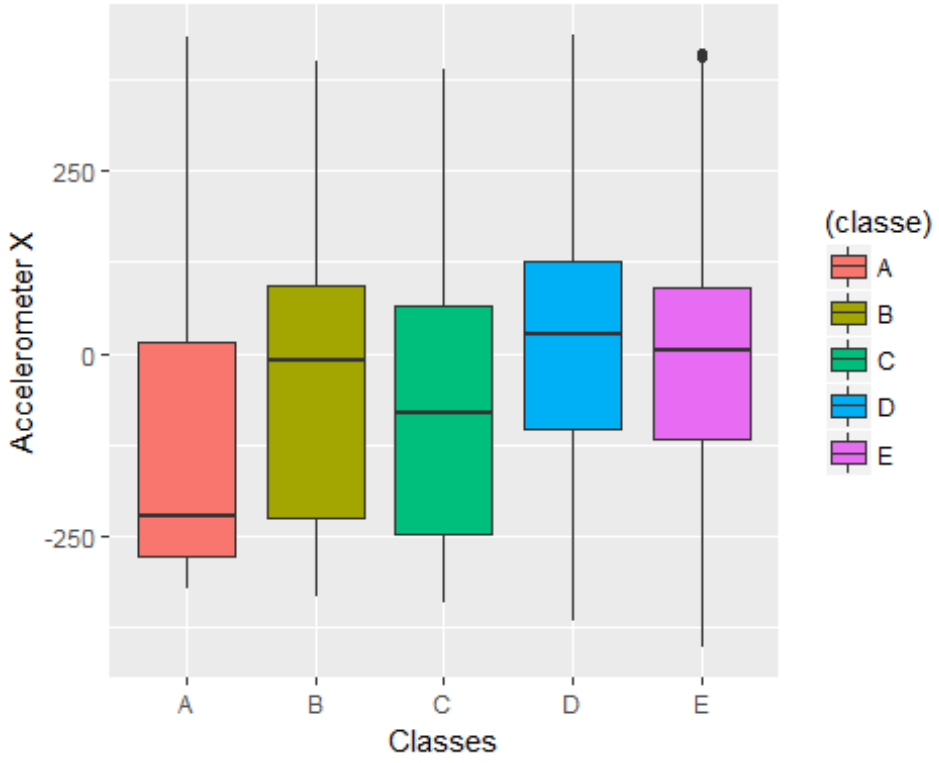
```
library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

library(caTools)

smp_size <- floor(0.75 * nrow(New_Training))
train_ind <- sample(seq_len(nrow(New_Training)), size = smp_size)
train <- New_Training[train_ind, ]
CV <- New_Training[-train_ind, ]

Model <- randomForest(formula,data=train)
pred_cv <- predict(Model,newdata=CV)
table(pred_cv,CV$classe)

##
## pred_cv    A    B    C    D    E
##       A 1370   37    1    6    6
##       B    4  867    9    1   18
##       C    1   10  795   20    7
##       D    6   12    4  776   13
##       E    0   13    9   11  910
```

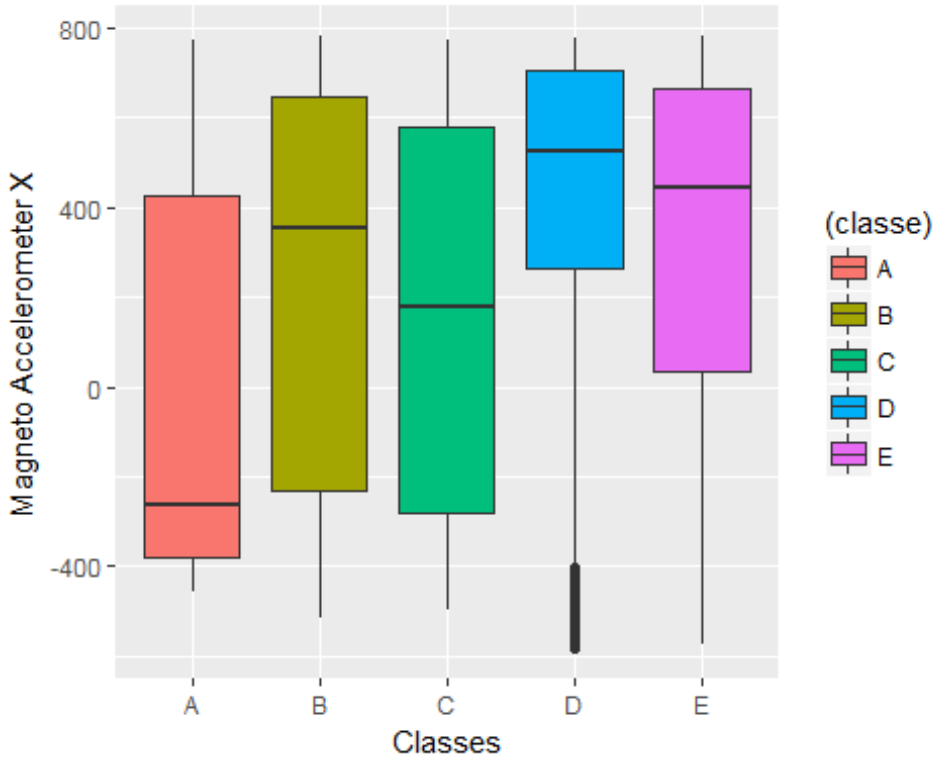
Largely, we are getting the classes right. The accuracy of class prediction is

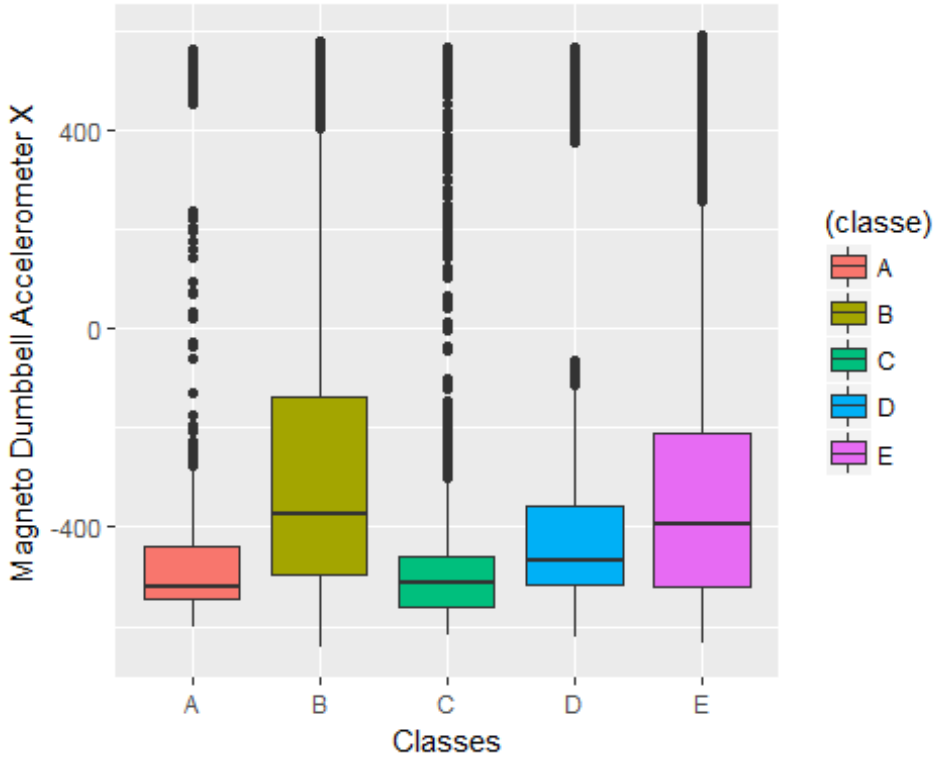
```
## [1] 0.9616796
```

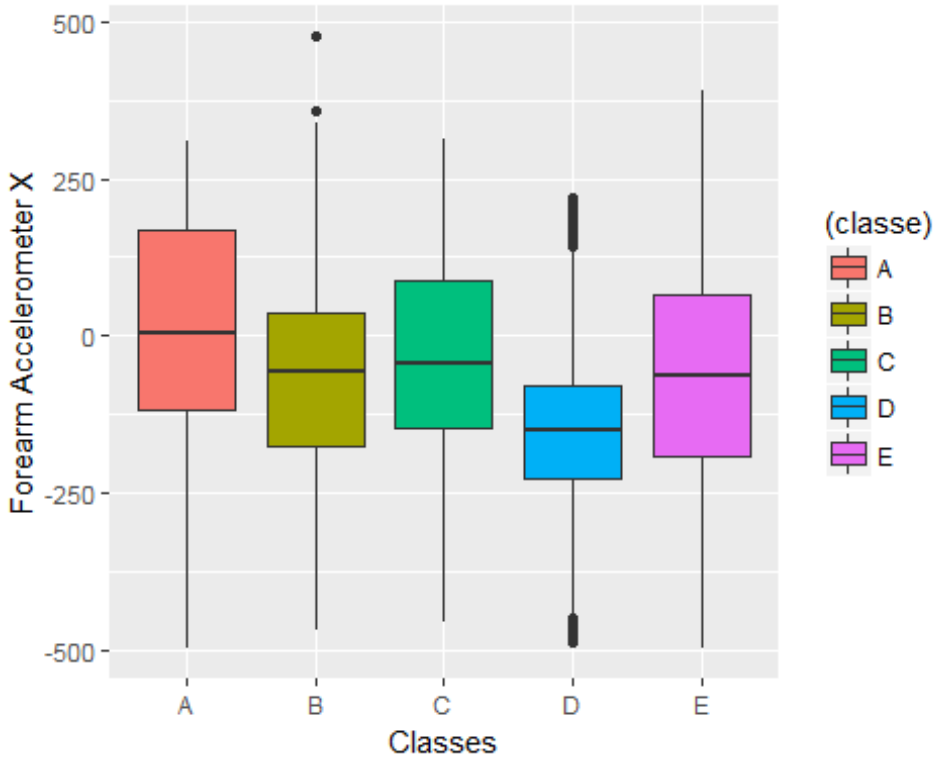
Appendix

The Plot of Different Accelerometer Data across X axix for the five classes.


250
0
-250
Accelerometer X
A
B
C
D
E
Classes
(classe)
A
B
C
D
E


800
400
0
-400
Magneto Accelerometer X
A
B
C
D
E
Classes
(classe)
A
B
C
D
E


Magneto Dumbbell Accelerometer X
400
0
-400
A
B
C
D
E
Classes
(classe)
A
B
C
D
E


Forearm Accelerometer X
500
250
0
-250
-500
A
B
C
D
E
Classes
(classe)
A
B
C
D
E