Human Activity Classification

January 09, 2021

1 Abstract

In recent years, there has been a significant increase in data measured from human activity/exercise thanks to the advent of wearable devices. Large amount of human activity data enables development of machine learning and deep learning to perform human activity prediction/classification, which would have potential impacts in improving human healthy and quality of life. In this report, we will build a machine learning model to classify how well the “Unilateral Dumbbell Biceps Curl” is performed using the “Weight Lifting Exercises Dataset”.

2 Getting data

Let’s download and read the data.

```
url_train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
fname_train <- "pml-training.csv"
download.file(url_train,fname_train)
df_train = read.csv(fname_train)

url_test <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
fname_test <- "pml-testing.csv"
download.file(url_test,fname_test)
df_test = read.csv(fname_test)
```

3 Exploratory Data Analysis

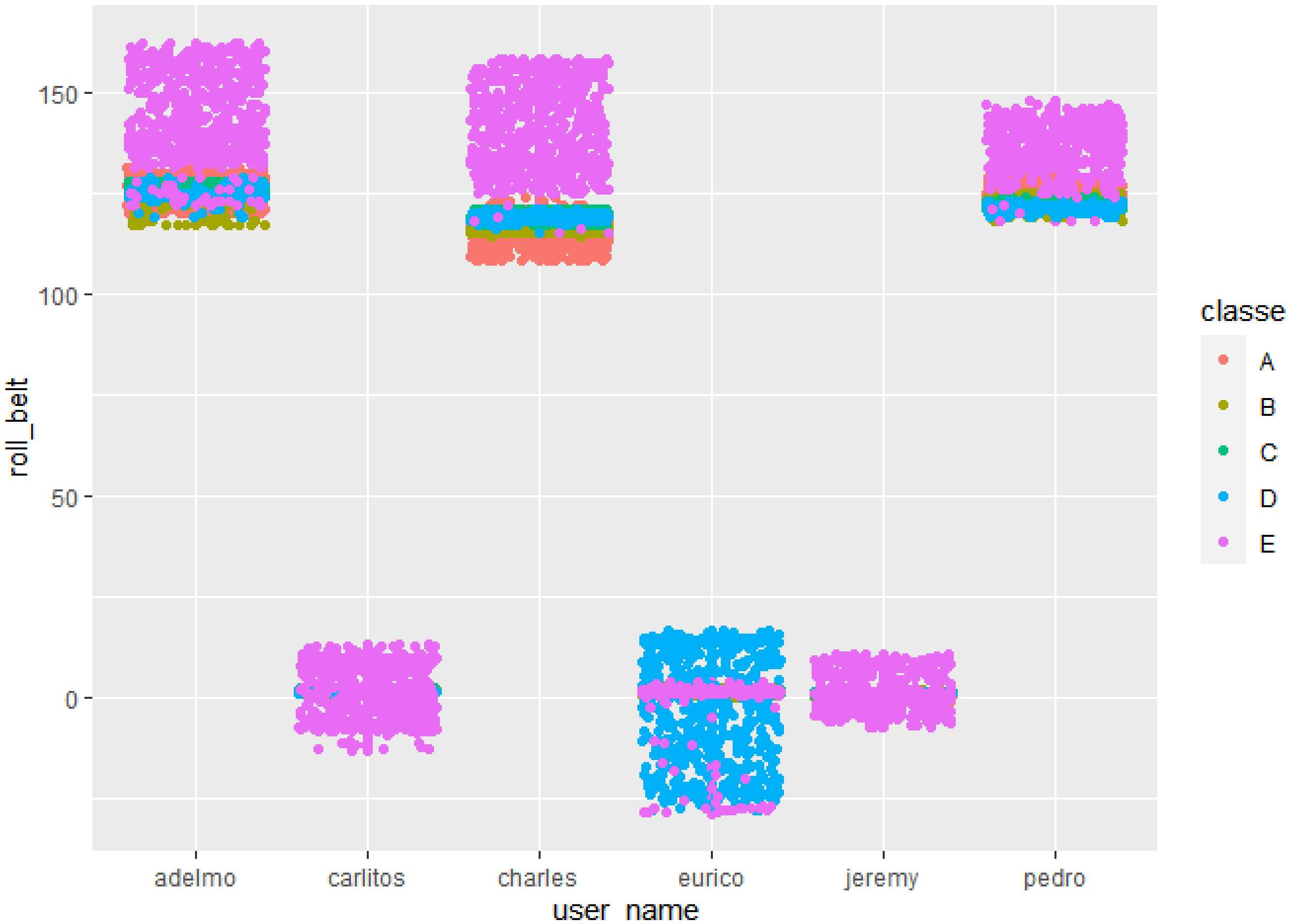
There are 5 class labels, which are A (i.e. correct movement) and B, C, D, and E (4 classes corresponding to common incorrect movements).

```
library(knitr)
kable(t(data.frame(labels = unique(df_train$classe))))
```

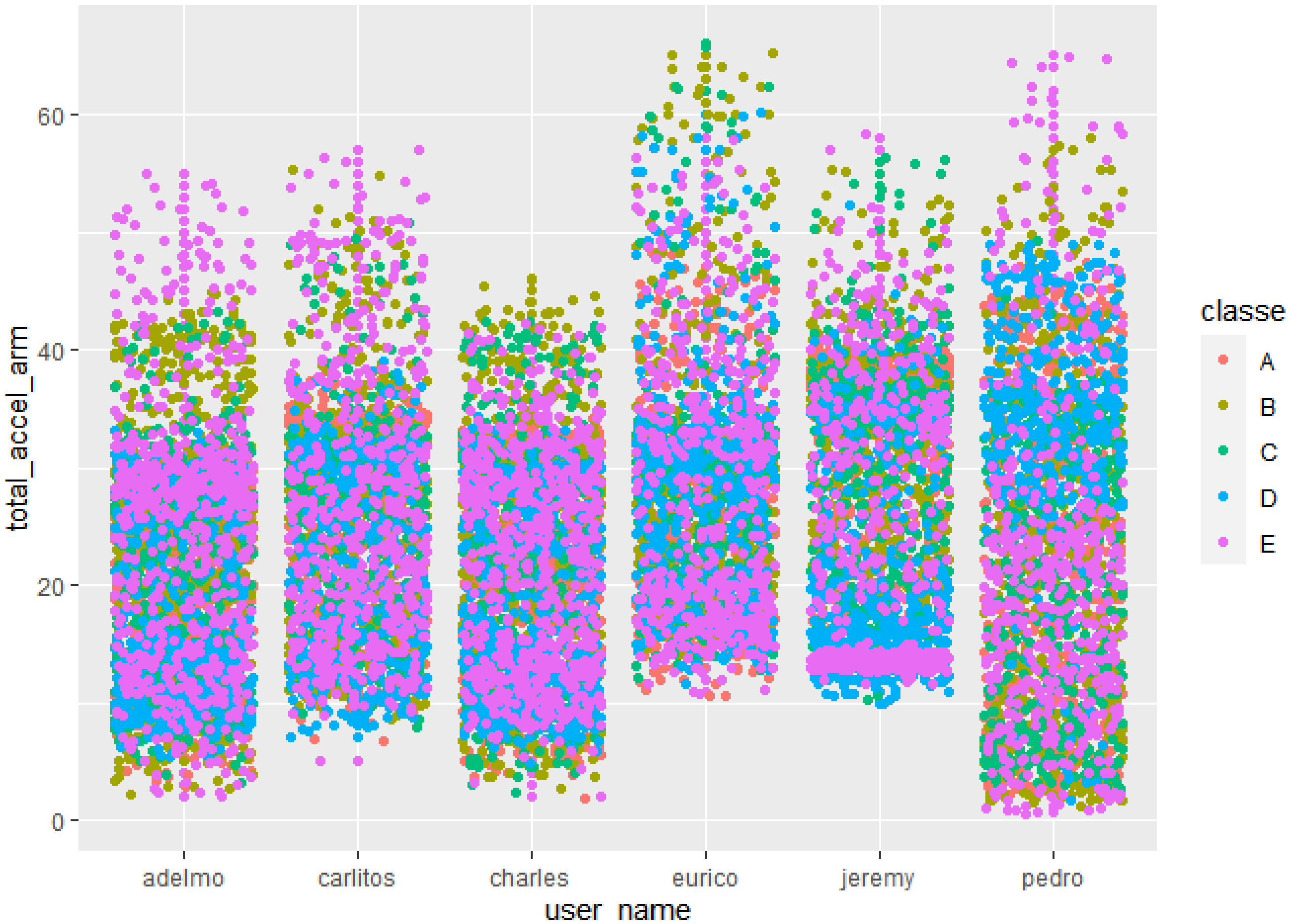
labels	A	B	C	D	E

Let’s check how the “roll_belt” predictor looks for each subject

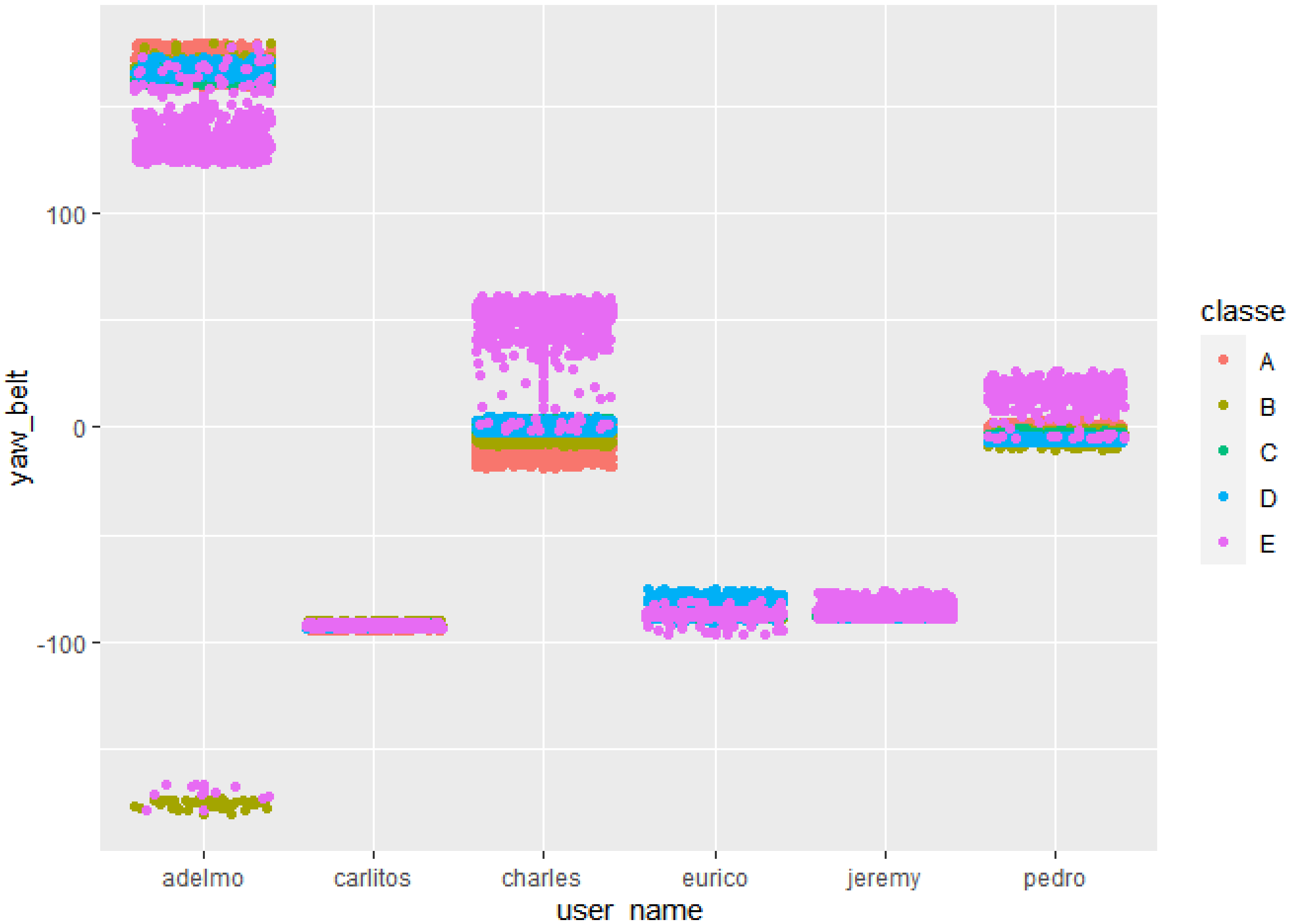
```
library(ggplot2)
ggplot(data = df_train, aes(x=user_name,y=roll_belt, colour=classe)) +
  geom_point() +
  geom_jitter()
```



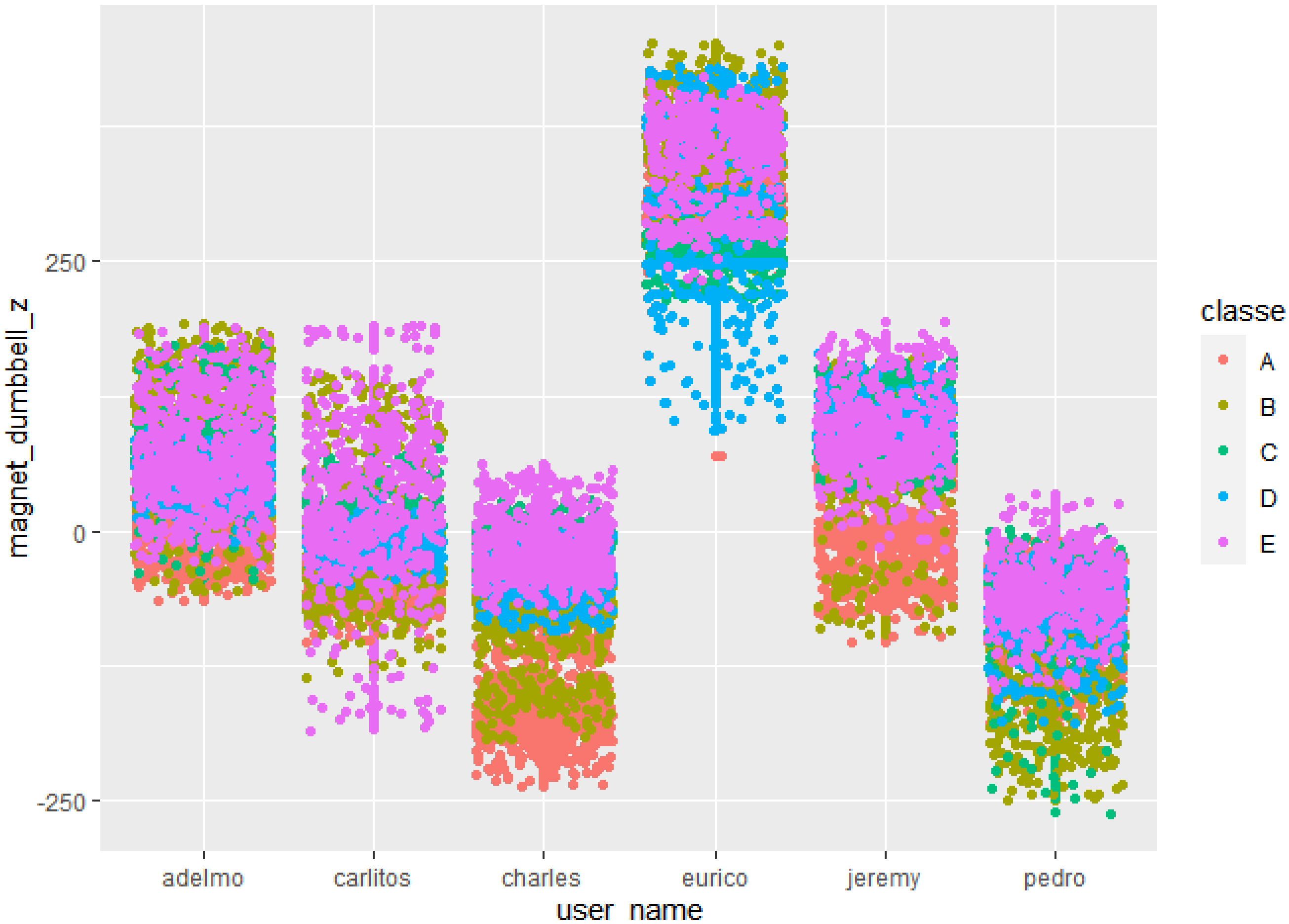

```
ggplot(data = df_train, aes(x=user_name,y=total_accel_arm, colour=classe)) +
  geom_point() +
  geom_jitter()
```



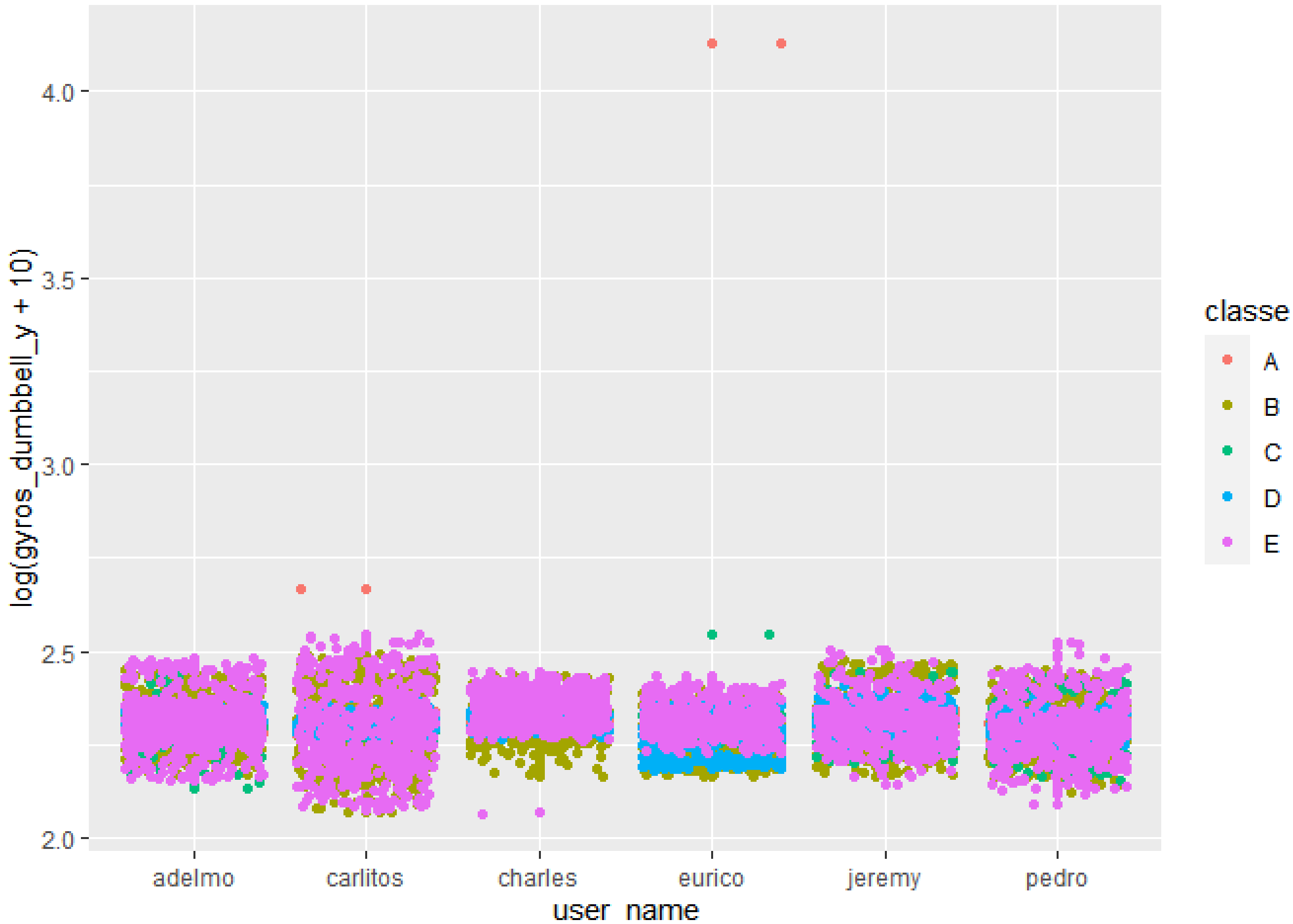

```
ggplot(data = df_train, aes(x=user_name,y=yaw_belt, colour=classe)) +
  geom_point() +
  geom_jitter()
```




```
ggplot(data = df_train, aes(x=user_name,y=magnet_dumbbell_z, colour=classe)) +
  geom_point() +
  geom_jitter()
```




```
ggplot(data = df_train, aes(x=user_name,y=log(gyros_dumbbell_y+10) , colour=classe)) +
  geom_point() +
  geom_jitter()
```



4 Feature Selections

Based on manual inspection and insights from the “Exploratory Data Analysis” section, we decided to include 52 numeric predictors as below.

```
suppressMessages(library(dplyr))
df_train1 <- select(df_train,
                    starts_with("total"),
                    starts_with("gyros"),
                    starts_with("accel"),
                    starts_with("magnet"),
                    starts_with("roll"),
                    starts_with("pitch"),
                    starts_with("yaw"),
                    starts_with("classe"))
df_test1 <- select(df_test,
                   starts_with("total"),
                   starts_with("accel"),
                   starts_with("magnet"),
                   starts_with("roll"),
                   starts_with("pitch"),
                   starts_with("yaw"),
                   starts_with("gyros"),)
```

5 Build Random Forest Model using 5-fold cross-validation

Let’s build a random forest classification model using caret package.

```
library(caret)
```

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

```
set.seed(12345)
trainctrl <- trainControl(method = "cv", number = 5, verboseIter = FALSE)
fit_final <- train(as.factor(classe) ~ .,
                data = df_train1,
                method = "rf",
                trControl=trainctrl,
                na.action=na.exclude)
```

Here is the summary of the trained random forest model.

```
print(fit_final)
```

```
## Random Forest
##
## 19622 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 15698, 15698, 15697, 15698, 15697
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9942921  0.9927795
##   27    0.9937825  0.9921349
##   52    0.9876668  0.9843970
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

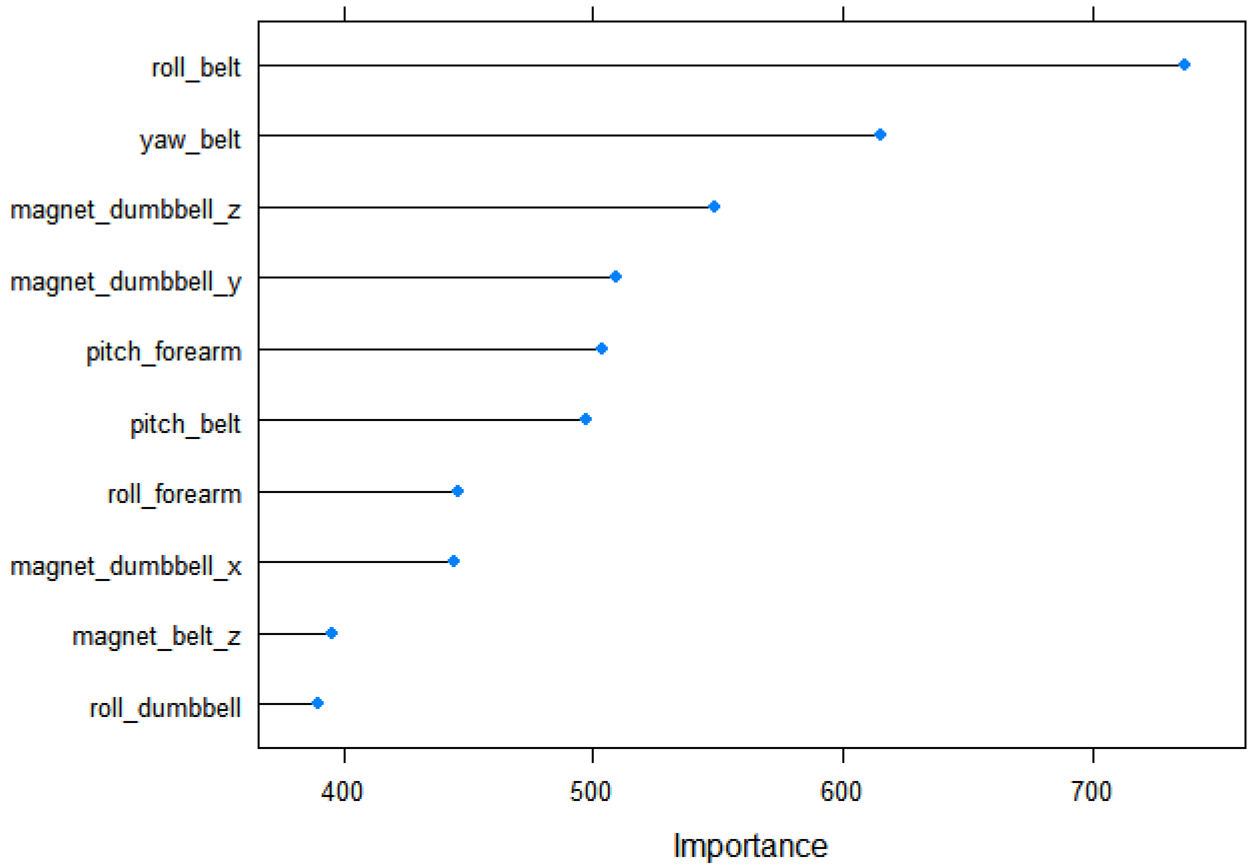
Here are list of features ordered based on their importance in classifying the 5 classes (A, B, C, D, E).

```
suppressMessages(library(caret))
varImp(fit_final, scale=FALSE)
```

```
## rf variable importance
## 
##   only 20 most important variables shown (out of 52)
## 
##                      Overall
## roll_belt              737.3
## yaw_belt               615.7
## magnet_dumbbell_z      548.4
## magnet_dumbbell_y      509.7
## pitch_forearm          503.6
## pitch_belt             497.5
## roll_forearm           445.9
## magnet_dumbbell_x      444.4
## magnet_belt_z          395.1
## roll_dumbbell          389.7
## accel_belt_z           383.6
## accel_dumbbell_y       382.6
## magnet_belt_y          359.4
## accel_dumbbell_z       353.4
## roll_arm               347.3
## accel_forearm_x        319.2
## gyros_belt_z           307.7
## yaw_dumbbell           303.5
## accel_dumbbell_x       298.9
## total_accel_dumbbell   295.1
```

Below is the plot of 5 predictors with highest importance.

```
plot(varImp(fit_final, scale=FALSE), top=10)
```



6 Predictions of the 20 test samples

```
predict(fit_final, df_test1)
```

```
##  [1] 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
```

7 Summary

We have build a machine learning model using random forest technique that intakes 52-predictors measured from wearable sensor to classify 5 classes. From the 5-fold cross-validation, the model achieves 99.43% accuracy.