Exercise quality prediction from accelerometers

Francisco Rodriguez

8 de noviembre de 2017

Summary

Using wearables devices it is now possible to collect a large amount of data about personal activity.

The goal of this project is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did an exercise.

Loading the data

Thee training data for this project are available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project can be found here.

```
setwd("C:/frodriguezp/DataScience/Rprojects");

urltrain<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
urltest<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
fnametrain<-"./data/pml-training.csv";
fnametest<-"./data/pml-testing.csv";

if(!file.exists(fnametrain))    download.file(urltrain, destfile = fnametrain);
if(!file.exists(fnametest))    download.file(urltest, destfile = fnametest);

mdtrain<-fread(fnametrain)
mdval<-fread(fnametest)
```

Preprocess

Data contains a lot of variables, but many of them contain missing values or *NA*. The first step will be to clean the data set in orther to include just the relevant parameters.
Also, the outcome is the *classe* variable. This variable is a factor with 5 levels *(A, B, C, D, E)*. Where each level corresponds to different ways of performing the exercise: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). To sum up, class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.
Columns form 1 to 7 are also remived because they do not contain usefull information for predicting the outcome.

```
mdtrain$classe<-factor(mdtrain$classe)

ic<-(colSums(is.na(mdtrain))>0) | (sapply(mdtrain,class) %in% "character");
ic[1:7]<-TRUE; ic<-which(ic)
mdtrain<-mdtrain[,-ic,with=FALSE]
```

Machine Learning

Cross Validation

In order to have an idea of the accuracy of the model (out of sample error), the *tran* data will be splitted into two sets: trainig and testing. This method will allow us to create the model just with the training data set and then use the model to compare the predictions of the testing set with the truth values.

To have even a better estimation of the accuracy, a k-fold cross validation method will be used. In R, these options can be controled using the `trainControl` command.

```
set.seed(333)
# Data Split
iTrain = createDataPartition(mdtrain$classe, p = 0.67)[[1]]
mdtest = mdtrain[-iTrain,]
mdtrain = mdtrain[iTrain,]

# Cross validation k-folds
train_control <- trainControl(method="cv", number=7)
```

Method

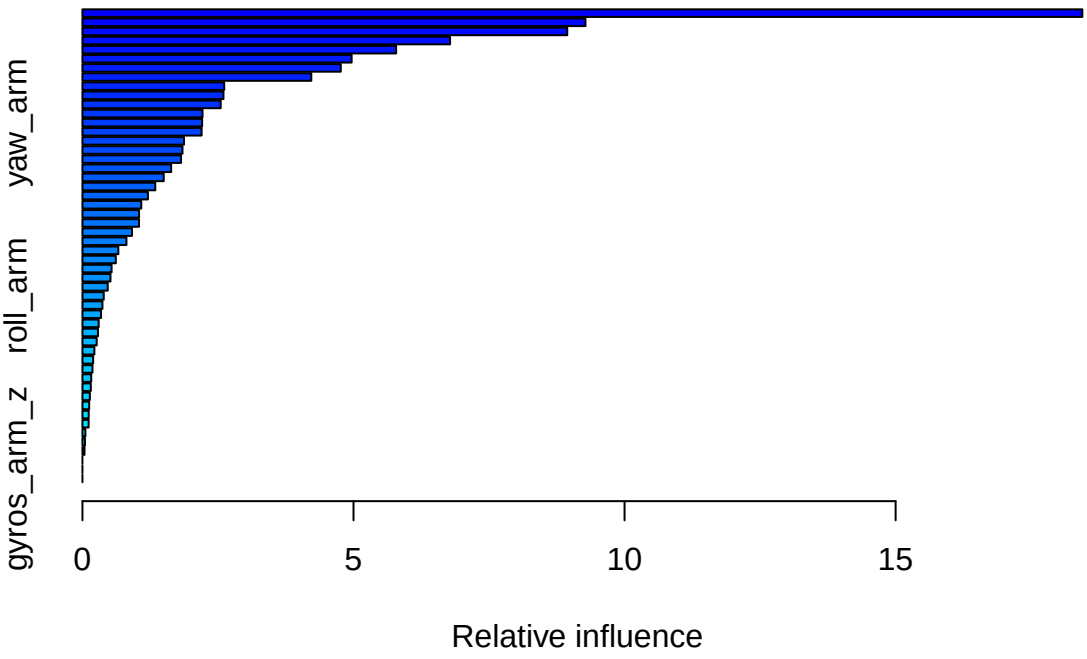
As this is a classifications problem; where we have 52 predictor variables and we have to predict the outcome (5-level factor), a **boosting** algorith will be used. The trees have to be deep enought to consider all the predictr variables, this can be checked with the `grid command`

```
gbmGrid <-  expand.grid(interaction.depth = floor(sqrt(NCOL(mdtrain))),
                        n.trees = (1:3)*50,
                        shrinkage = 0.1,
                        n.minobsinnode = 10)
gbmmodel<-train(classe ~ ., data=mdtrain, method="gbm",
                verbose=FALSE,
                preProc = c("center", "scale"),
                trControl=train_control,
                tuneGrid = gbmGrid )
gbmres<-gbmmodel$results[3,c(2,4,5,7)]; row.names(gbmres)<-NULL
kable(gbmres)
```

interaction.depth	n.trees	Accuracy	AccuracySD
7	150	0.9879829	0.0019891

The in-sample accuracy is 0.9879829, which is good enough.
The main predictor variables are shown in the following table.

```
rivar<-head(summary(gbmmodel$finalModel),7); row.names(rivar)<-NULL
```




```
names(rivar)<-c("variable", "relative_influence")
kable(rivar)
```

variable	relative_influence
roll_belt	18.445024
pitch_forearm	9.278833
yaw_belt	8.942635
magnet_dumbbell_z	6.781294
magnet_dumbbell_y	5.786648
pitch_belt	4.965827
roll_forearm	4.765723

Results

One of the bests ways to obtain the out-sample error is the *Confussion Matrix*, this command will compare the truth values of the test set (that have not been used for the model) with the predictions.

```
cm<-confusionMatrix(mdtest$classe, predict(gbmmodel,mdtest)); cm
```

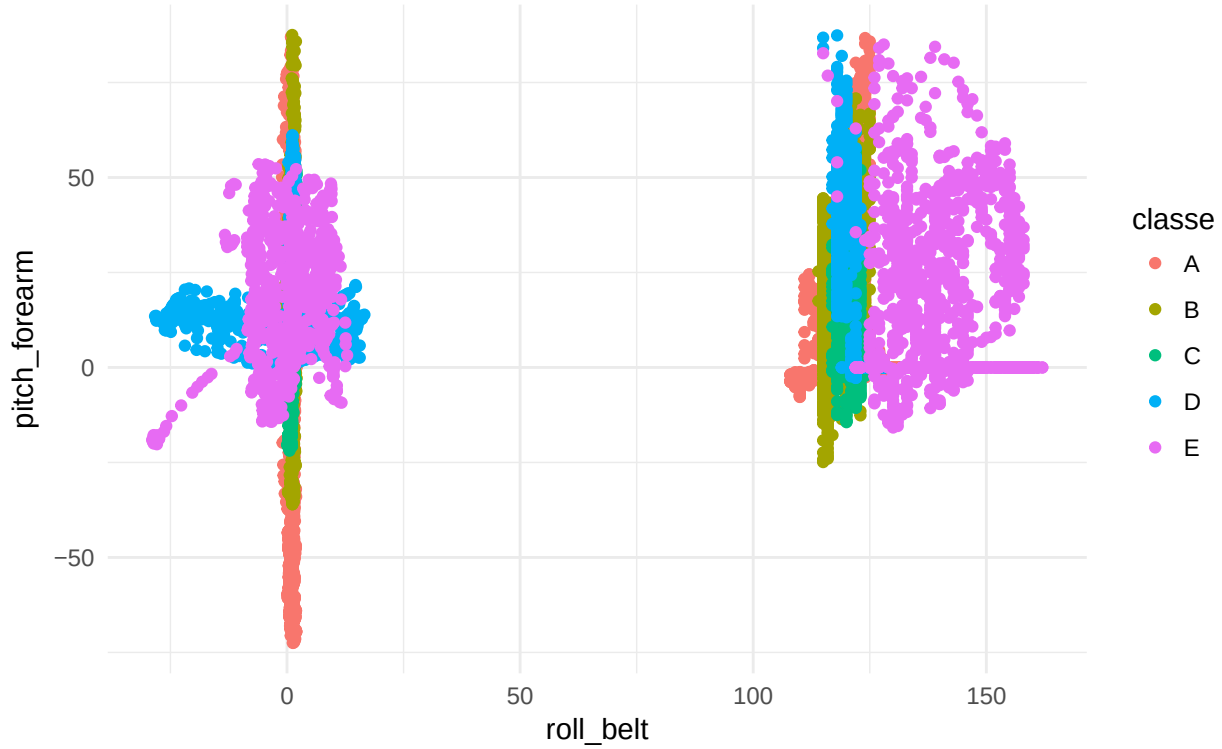
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1836    4    1    0    0
##          B    7 1233   13    0    0
```

```
##          C    0    6 1118    5    0
##          D    0    0   28 1029    4
##          E    0    2    6    5 1177
##
## Overall Statistics
##
##                Accuracy : 0.9875
##                  95% CI : (0.9845, 0.9901)
##     No Information Rate : 0.2847
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9842
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9962   0.9904   0.9588   0.9904   0.9966
## Specificity            0.9989   0.9962   0.9979   0.9941   0.9975
## Pos Pred Value         0.9973   0.9840   0.9903   0.9698   0.9891
## Neg Pred Value         0.9985   0.9977   0.9910   0.9982   0.9992
## Prevalence             0.2847   0.1923   0.1801   0.1605   0.1824
## Detection Rate         0.2836   0.1905   0.1727   0.1589   0.1818
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9976   0.9933   0.9784   0.9922   0.9971
```

The test accuracy is 0.9874884, so we can conclude that the model estimates the *quality* of the exercise really well.

To have an idea of the model, the following plot show a sccated plot with the two main predictors colored by the outcome.

```
ggplot(mdtrain, aes(x=roll_belt, y=pitch_forearm, colour=classe)) +
    geom_point() +
    theme_minimal()
```



The plot shows that the different classes are classified in different *clusters*.

Prediction

Finally, the model will be used to predict the *quality* of the exercise to the given test data, where the outcome variable is unknown.

```
kable(data.frame(id=mdval$problem_id, name=mdval$user_name, classe=predict(gbmmodel,mdval)))
```

id	name	classe
1	pedro	B
2	jeremy	A
3	jeremy	B
4	adelmo	A
5	eurico	A
6	jeremy	E
7	jeremy	D
8	jeremy	B
9	carlitos	A
10	charles	A
11	carlitos	B
12	jeremy	C
13	eurico	B
14	jeremy	A
15	jeremy	E
16	eurico	E
17	pedro	A
18	carlitos	B

id	name	classe
19	pedro	B
20	eurico	B

Conclusion

From a data set with 52 predictor variables (from accelerometers on the belt, forearm, arm, and dumbell) it has been possible to train a model to predict the manner in which a subject was peforming an exercise.

The model has obteined really good out-sample errors, the error matrix from the test data set can be seen in the following table.

```
kable(cm$table)
```

	A	B	C	D	E
A	1836	4	1	0	0
B	7	1233	13	0	0
C	0	6	1118	5	0
D	0	0	28	1029	4
E	0	2	6	5	1177

Where the test data set has been obtained splitting the original training set.