

Qualitative Activity Recognition of Weight Lifting Exercises

Frank Inklaar

10 februari 2019

Synopsis

In this assignment we aim to predict whether a physical exercise (unilateral dumbel lift) has been performed correctly or whether a well know mistake has been made. We have a training set created by recording the activities of 6 individuals each performing the exercise correctly and performing the exercise with 4 known mistakes. Each recording has been labeled with a 'class' variable where 'A' is the correct way of performing the exercise and 'B' through 'E' are the erroneous ways. Each exercise was recorded by 4 sensors attached to upper arm, wrist, belly and dumbel respectively. Each sensor is measuring acceleration in three dimensions as well as rotation in three dimensional axes. We aim to train a model that can predict from each measurement to what 'classe' it belongs. With this model we would be able to provide real time feedback to the individual performing the exercise.

Data processing

Loading the data

We load the data. We suppress the immediate conversion to factors as it seems to have some unwanted effect on this dataset. We'll convert the right columns to factor variables later on.

```
set.seed(12345)
training<-read.csv("pml-training.csv",stringsAsFactors = FALSE)
testing<-read.csv("pml-testing.csv",stringsAsFactors = FALSE)
```

Exploring the data

As a first step, let's get some very basic info about the dataset we've just read

```
print(dim(training))
```

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

```
head(training)
```

```
##   X user_name raw_timestamp_part_1 raw_timestamp_part_2  cvtd_timestamp
## 1 1  carlitos          1323084231          788290 05/12/2011 11:23
## 2 2  carlitos          1323084231          808298 05/12/2011 11:23
## 3 3  carlitos          1323084231          820366 05/12/2011 11:23
## 4 4  carlitos          1323084232          120339 05/12/2011 11:23
## 5 5  carlitos          1323084232          196328 05/12/2011 11:23
## 6 6  carlitos          1323084232          304277 05/12/2011 11:23
##   new_window num_window roll_belt pitch_belt yaw_belt total_accel_belt
## 1         no          11      1.41      8.07    -94.4              3
## 2         no          11      1.41      8.07    -94.4              3
```

```

## 3      no      11      1.42      8.07      -94.4      3
## 4      no      12      1.48      8.05      -94.4      3
## 5      no      12      1.48      8.07      -94.4      3
## 6      no      12      1.45      8.06      -94.4      3
##      kurtosis_roll_belt kurtosis_pitch_belt kurtosis_yaw_belt
## 1
## 2
## 3
## 4
## 5
## 6
##      skewness_roll_belt skewness_roll_belt.1 skewness_yaw_belt max_roll_belt
## 1                                          NA
## 2                                          NA
## 3                                          NA
## 4                                          NA
## 5                                          NA
## 6                                          NA
##      max_pitch_belt max_yaw_belt min_roll_belt min_pitch_belt min_yaw_belt
## 1      NA      NA      NA      NA
## 2      NA      NA      NA      NA
## 3      NA      NA      NA      NA
## 4      NA      NA      NA      NA
## 5      NA      NA      NA      NA
## 6      NA      NA      NA      NA
##      amplitude_roll_belt amplitude_pitch_belt amplitude_yaw_belt
## 1      NA      NA
## 2      NA      NA
## 3      NA      NA
## 4      NA      NA
## 5      NA      NA
## 6      NA      NA
##      var_total_accel_belt avg_roll_belt stddev_roll_belt var_roll_belt
## 1      NA      NA      NA      NA
## 2      NA      NA      NA      NA
## 3      NA      NA      NA      NA
## 4      NA      NA      NA      NA
## 5      NA      NA      NA      NA
## 6      NA      NA      NA      NA
##      avg_pitch_belt stddev_pitch_belt var_pitch_belt avg_yaw_belt
## 1      NA      NA      NA      NA
## 2      NA      NA      NA      NA
## 3      NA      NA      NA      NA
## 4      NA      NA      NA      NA
## 5      NA      NA      NA      NA
## 6      NA      NA      NA      NA
##      stddev_yaw_belt var_yaw_belt gyros_belt_x gyros_belt_y gyros_belt_z
## 1      NA      NA      0.00      0.00      -0.02
## 2      NA      NA      0.02      0.00      -0.02
## 3      NA      NA      0.00      0.00      -0.02
## 4      NA      NA      0.02      0.00      -0.03
## 5      NA      NA      0.02      0.02      -0.02
## 6      NA      NA      0.02      0.00      -0.02
##      accel_belt_x accel_belt_y accel_belt_z magnet_belt_x magnet_belt_y

```

## 1	-21	4	22	-3	599	
## 2	-22	4	22	-7	608	
## 3	-20	5	23	-2	600	
## 4	-22	3	21	-6	604	
## 5	-21	2	24	-6	600	
## 6	-21	4	21	0	603	
##	magnet_belt_z	roll_arm	pitch_arm	yaw_arm	total_accel_arm	var_accel_arm
## 1	-313	-128	22.5	-161	34	NA
## 2	-311	-128	22.5	-161	34	NA
## 3	-305	-128	22.5	-161	34	NA
## 4	-310	-128	22.1	-161	34	NA
## 5	-302	-128	22.1	-161	34	NA
## 6	-312	-128	22.0	-161	34	NA
##	avg_roll_arm	stddev_roll_arm	var_roll_arm	avg_pitch_arm	stddev_pitch_arm	
## 1	NA	NA	NA	NA	NA	NA
## 2	NA	NA	NA	NA	NA	NA
## 3	NA	NA	NA	NA	NA	NA
## 4	NA	NA	NA	NA	NA	NA
## 5	NA	NA	NA	NA	NA	NA
## 6	NA	NA	NA	NA	NA	NA
##	var_pitch_arm	avg_yaw_arm	stddev_yaw_arm	var_yaw_arm	gyros_arm_x	
## 1	NA	NA	NA	NA	0.00	
## 2	NA	NA	NA	NA	0.02	
## 3	NA	NA	NA	NA	0.02	
## 4	NA	NA	NA	NA	0.02	
## 5	NA	NA	NA	NA	0.00	
## 6	NA	NA	NA	NA	0.02	
##	gyros_arm_y	gyros_arm_z	accel_arm_x	accel_arm_y	accel_arm_z	magnet_arm_x
## 1	0.00	-0.02	-288	109	-123	-368
## 2	-0.02	-0.02	-290	110	-125	-369
## 3	-0.02	-0.02	-289	110	-126	-368
## 4	-0.03	0.02	-289	111	-123	-372
## 5	-0.03	0.00	-289	111	-123	-374
## 6	-0.03	0.00	-289	111	-122	-369
##	magnet_arm_y	magnet_arm_z	kurtosis_roll_arm	kurtosis_pitch_arm		
## 1	337	516				
## 2	337	513				
## 3	344	513				
## 4	344	512				
## 5	337	506				
## 6	342	513				
##	kurtosis_yaw_arm	skewness_roll_arm	skewness_pitch_arm	skewness_yaw_arm		
## 1						
## 2						
## 3						
## 4						
## 5						
## 6						
##	max_roll_arm	max_pitch_arm	max_yaw_arm	min_roll_arm	min_pitch_arm	
## 1	NA	NA	NA	NA	NA	
## 2	NA	NA	NA	NA	NA	
## 3	NA	NA	NA	NA	NA	
## 4	NA	NA	NA	NA	NA	
## 5	NA	NA	NA	NA	NA	

## 6	NA	NA	NA	NA	NA
##	min_yaw_arm	amplitude_roll_arm	amplitude_pitch_arm	amplitude_yaw_arm	
## 1	NA	NA	NA	NA	NA
## 2	NA	NA	NA	NA	NA
## 3	NA	NA	NA	NA	NA
## 4	NA	NA	NA	NA	NA
## 5	NA	NA	NA	NA	NA
## 6	NA	NA	NA	NA	NA
##	roll_dumbbell	pitch_dumbbell	yaw_dumbbell	kurtosis_roll_dumbbell	
## 1	13.05217	-70.49400	-84.87394		
## 2	13.13074	-70.63751	-84.71065		
## 3	12.85075	-70.27812	-85.14078		
## 4	13.43120	-70.39379	-84.87363		
## 5	13.37872	-70.42856	-84.85306		
## 6	13.38246	-70.81759	-84.46500		
##	kurtosis_pitch_dumbbell	kurtosis_yaw_dumbbell	skewness_roll_dumbbell		
## 1					
## 2					
## 3					
## 4					
## 5					
## 6					
##	skewness_pitch_dumbbell	skewness_yaw_dumbbell	max_roll_dumbbell		
## 1				NA	
## 2				NA	
## 3				NA	
## 4				NA	
## 5				NA	
## 6				NA	
##	max_pitch_dumbbell	max_yaw_dumbbell	min_roll_dumbbell	min_pitch_dumbbell	
## 1	NA		NA	NA	
## 2	NA		NA	NA	
## 3	NA		NA	NA	
## 4	NA		NA	NA	
## 5	NA		NA	NA	
## 6	NA		NA	NA	
##	min_yaw_dumbbell	amplitude_roll_dumbbell	amplitude_pitch_dumbbell		
## 1		NA	NA		
## 2		NA	NA		
## 3		NA	NA		
## 4		NA	NA		
## 5		NA	NA		
## 6		NA	NA		
##	amplitude_yaw_dumbbell	total_accel_dumbbell	var_accel_dumbbell		
## 1		37	NA		
## 2		37	NA		
## 3		37	NA		
## 4		37	NA		
## 5		37	NA		
## 6		37	NA		
##	avg_roll_dumbbell	stddev_roll_dumbbell	var_roll_dumbbell		
## 1	NA	NA	NA		
## 2	NA	NA	NA		
## 3	NA	NA	NA		

## 4	NA	NA	NA	
## 5	NA	NA	NA	
## 6	NA	NA	NA	
##	avg_pitch_dumbbell	stddev_pitch_dumbbell	var_pitch_dumbbell	
## 1	NA	NA	NA	
## 2	NA	NA	NA	
## 3	NA	NA	NA	
## 4	NA	NA	NA	
## 5	NA	NA	NA	
## 6	NA	NA	NA	
##	avg_yaw_dumbbell	stddev_yaw_dumbbell	var_yaw_dumbbell	gyros_dumbbell_x
## 1	NA	NA	NA	0
## 2	NA	NA	NA	0
## 3	NA	NA	NA	0
## 4	NA	NA	NA	0
## 5	NA	NA	NA	0
## 6	NA	NA	NA	0
##	gyros_dumbbell_y	gyros_dumbbell_z	accel_dumbbell_x	accel_dumbbell_y
## 1	-0.02	0.00	-234	47
## 2	-0.02	0.00	-233	47
## 3	-0.02	0.00	-232	46
## 4	-0.02	-0.02	-232	48
## 5	-0.02	0.00	-233	48
## 6	-0.02	0.00	-234	48
##	accel_dumbbell_z	magnet_dumbbell_x	magnet_dumbbell_y	magnet_dumbbell_z
## 1	-271	-559	293	-65
## 2	-269	-555	296	-64
## 3	-270	-561	298	-63
## 4	-269	-552	303	-60
## 5	-270	-554	292	-68
## 6	-269	-558	294	-66
##	roll_forearm	pitch_forearm	yaw_forearm	kurtosis_roll_forearm
## 1	28.4	-63.9	-153	
## 2	28.3	-63.9	-153	
## 3	28.3	-63.9	-152	
## 4	28.1	-63.9	-152	
## 5	28.0	-63.9	-152	
## 6	27.9	-63.9	-152	
##	kurtosis_pitch_forearm	kurtosis_yaw_forearm	skewness_roll_forearm	
## 1				
## 2				
## 3				
## 4				
## 5				
## 6				
##	skewness_pitch_forearm	skewness_yaw_forearm	max_roll_forearm	
## 1			NA	
## 2			NA	
## 3			NA	
## 4			NA	
## 5			NA	
## 6			NA	
##	max_pitch_forearm	max_yaw_forearm	min_roll_forearm	min_pitch_forearm
## 1	NA		NA	NA

## 2	NA		NA	NA
## 3	NA		NA	NA
## 4	NA		NA	NA
## 5	NA		NA	NA
## 6	NA		NA	NA
##	min_yaw_forearm	amplitude_roll_forearm	amplitude_pitch_forearm	
## 1		NA	NA	
## 2		NA	NA	
## 3		NA	NA	
## 4		NA	NA	
## 5		NA	NA	
## 6		NA	NA	
##	amplitude_yaw_forearm	total_accel_forearm	var_accel_forearm	
## 1		36	NA	
## 2		36	NA	
## 3		36	NA	
## 4		36	NA	
## 5		36	NA	
## 6		36	NA	
##	avg_roll_forearm	stddev_roll_forearm	var_roll_forearm	avg_pitch_forearm
## 1	NA	NA	NA	NA
## 2	NA	NA	NA	NA
## 3	NA	NA	NA	NA
## 4	NA	NA	NA	NA
## 5	NA	NA	NA	NA
## 6	NA	NA	NA	NA
##	stddev_pitch_forearm	var_pitch_forearm	avg_yaw_forearm	
## 1	NA	NA	NA	
## 2	NA	NA	NA	
## 3	NA	NA	NA	
## 4	NA	NA	NA	
## 5	NA	NA	NA	
## 6	NA	NA	NA	
##	stddev_yaw_forearm	var_yaw_forearm	gyros_forearm_x	gyros_forearm_y
## 1	NA	NA	0.03	0.00
## 2	NA	NA	0.02	0.00
## 3	NA	NA	0.03	-0.02
## 4	NA	NA	0.02	-0.02
## 5	NA	NA	0.02	0.00
## 6	NA	NA	0.02	-0.02
##	gyros_forearm_z	accel_forearm_x	accel_forearm_y	accel_forearm_z
## 1	-0.02	192	203	-215
## 2	-0.02	192	203	-216
## 3	0.00	196	204	-213
## 4	0.00	189	206	-214
## 5	-0.02	189	206	-214
## 6	-0.03	193	203	-215
##	magnet_forearm_x	magnet_forearm_y	magnet_forearm_z	classe
## 1	-17	654	476	A
## 2	-18	661	473	A
## 3	-18	658	469	A
## 4	-16	658	469	A
## 5	-17	655	473	A
## 6	-9	660	478	A

From the first check it's obvious that not all columns seem to contain data. Further inspection reveals that the dataset contains two distinct kind of data. The records labeled `new_window = 'no'` contain raw data from the sensors, while the records labeled `'yes'` also contain statistical derivatives from one complete excersize.

```
# check how the statistical columns relate tot the 'new_window' variable
print(table(training$new_window,training$kurtosis_roll_belt==""))
```

```
##
##      FALSE  TRUE
##   no      0 19216
##   yes    406     0
```

```
# check if the test set contains any 'new_window' rows
print(table(testing$new_window))
```

```
##
## no
## 20
```

Since the test set contains only records where `new_window = 'no'` we cannot use this information in our prediction. For this reason we will get rid of the columns that only have information for the `'yes'` records. We will also get rid of the timestamp information, as this would result in recognizing `'when'` the activity was performed rather than `'how'`.

Cleaning the Data

Since the test set has only `new_window = "no"` records we can kick the `"yes"` records out of the training set and after that, get rid of all the columns that have a constant value or all NA's

```
training<-training[training$new_window=="no",]

na_cols<-which(colSums(is.na(training))==nrow(training))

blank_cols<-which(colSums(training[,]=="" )==nrow(training))

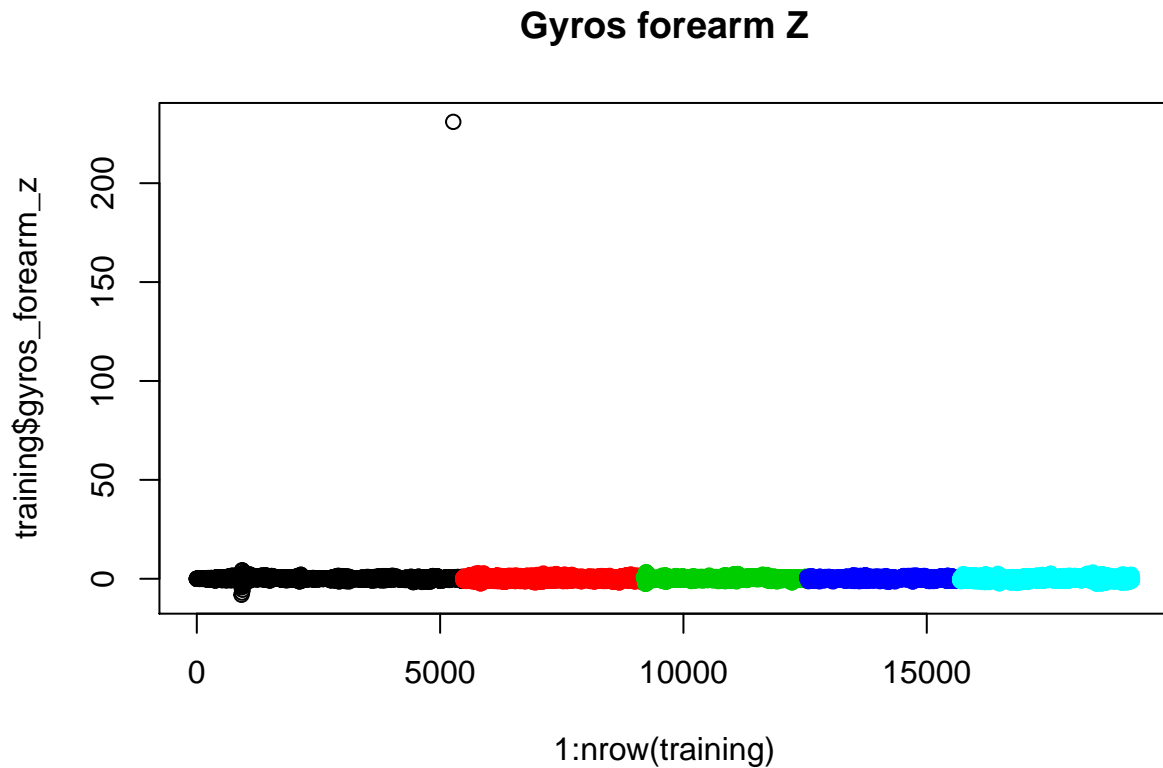
training<-training[,c(-na_cols,-blank_cols)]
testing<-testing[,c(-na_cols,-blank_cols)]

#convert uses_name and classe to factor variables
training$user_name<-factor(training$user_name)
testing$user_name<-factor(testing$user_name,levels=levels(training$user_name))
training$classe<-factor(training$classe)

#get rid of all timestamp related columns
training$X<-NULL; testing$X<-NULL
training$raw_timestamp_part_1<-NULL; testing$raw_timestamp_part_1<-NULL
training$raw_timestamp_part_2<-NULL; testing$raw_timestamp_part_2<-NULL
training$cvtd_timestamp<-NULL; testing$cvtd_timestamp<-NULL
training$new_window<-NULL; testing$new_window<-NULL
training$num_window<-NULL; testing$num_window<-NULL
#Filter(function(x) length(unique(x))==1, training)
```

Finally, exploratory data analyses shows that there are some outliers. For instance if we look at the `gyros_forearm_z` column:

```
plot(1:nrow(training),training$gyros_forearm_z,col=training$classe); title("Gyros forearm Z")
```



There is a single outlier with a value above 300. We'll delete that one from the training set

```
which(training$gyros_forearm_z>200)
```

```
## [1] 5270
```

```
#record 5270 is an outlier, get rid of it  
training<-training[-5270,]
```

Training the model

Model selection

From the nature of this experiment, it seems probable that the classification result will not have a linear dependency of the sensor outputs. For this case a Random Forest seem to be a proper model. We'll train a Random Forest model with 5 fold cross validation to get a predicted value for our out of set prediction error.

Training the model

First we load the necessary libraries and set up the laptop for multithreading to reduce run time.

```
library(caret)
```



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

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

```
library(parallel)
```

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
library(beep)
```

```
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
```

```
registerDoParallel(cluster)
```

```
head(training)
```

```
##   user_name roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x
## 1 carlitos    1.41    8.07   -94.4          3          0.00
## 2 carlitos    1.41    8.07   -94.4          3          0.02
## 3 carlitos    1.42    8.07   -94.4          3          0.00
## 4 carlitos    1.48    8.05   -94.4          3          0.02
## 5 carlitos    1.48    8.07   -94.4          3          0.02
## 6 carlitos    1.45    8.06   -94.4          3          0.02
##   gyros_belt_y gyros_belt_z accel_belt_x accel_belt_y accel_belt_z
## 1          0.00        -0.02         -21           4          22
## 2          0.00        -0.02         -22           4          22
## 3          0.00        -0.02         -20           5          23
## 4          0.00        -0.03         -22           3          21
## 5          0.02        -0.02         -21           2          24
## 6          0.00        -0.02         -21           4          21
##   magnet_belt_x magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm
## 1           -3          599         -313     -128    22.5   -161
## 2           -7          608         -311     -128    22.5   -161
## 3           -2          600         -305     -128    22.5   -161
## 4           -6          604         -310     -128    22.1   -161
## 5           -6          600         -302     -128    22.1   -161
## 6            0          603         -312     -128    22.0   -161
##   total_accel_arm gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x
## 1             34          0.00          0.00        -0.02        -288
## 2             34          0.02         -0.02        -0.02        -290
## 3             34          0.02         -0.02        -0.02        -289
## 4             34          0.02         -0.03          0.02        -289
## 5             34          0.00         -0.03          0.00        -289
## 6             34          0.02         -0.03          0.00        -289
##   accel_arm_y accel_arm_z magnet_arm_x magnet_arm_y magnet_arm_z
## 1          109        -123        -368          337          516
## 2          110        -125        -369          337          513
## 3          110        -126        -368          344          513
## 4          111        -123        -372          344          512
## 5          111        -123        -374          337          506
## 6          111        -122        -369          342          513
##   roll_dumbbell pitch_dumbbell yaw_dumbbell total_accel_dumbbell
## 1    13.05217    -70.49400    -84.87394          37
## 2    13.13074    -70.63751    -84.71065          37
## 3    12.85075    -70.27812    -85.14078          37
## 4    13.43120    -70.39379    -84.87363          37
```

```

## 5      13.37872      -70.42856      -84.85306              37
## 6      13.38246      -70.81759      -84.46500              37
##      gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x
## 1              0              -0.02              0.00             -234
## 2              0              -0.02              0.00             -233
## 3              0              -0.02              0.00             -232
## 4              0              -0.02             -0.02             -232
## 5              0              -0.02              0.00             -233
## 6              0              -0.02              0.00             -234
##      accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y
## 1              47             -271             -559              293
## 2              47             -269             -555              296
## 3              46             -270             -561              298
## 4              48             -269             -552              303
## 5              48             -270             -554              292
## 6              48             -269             -558              294
##      magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm
## 1             -65             28.4             -63.9             -153
## 2             -64             28.3             -63.9             -153
## 3             -63             28.3             -63.9             -152
## 4             -60             28.1             -63.9             -152
## 5             -68             28.0             -63.9             -152
## 6             -66             27.9             -63.9             -152
##      total_accel_forearm gyros_forearm_x gyros_forearm_y gyros_forearm_z
## 1              36              0.03              0.00             -0.02
## 2              36              0.02              0.00             -0.02
## 3              36              0.03             -0.02              0.00
## 4              36              0.02             -0.02              0.00
## 5              36              0.02              0.00             -0.02
## 6              36              0.02             -0.02             -0.03
##      accel_forearm_x accel_forearm_y accel_forearm_z magnet_forearm_x
## 1              192              203             -215              -17
## 2              192              203             -216              -18
## 3              196              204             -213              -18
## 4              189              206             -214              -16
## 5              189              206             -214              -17
## 6              193              203             -215              -9
##      magnet_forearm_y magnet_forearm_z classe
## 1              654              476         A
## 2              661              473         A
## 3              658              469         A
## 4              658              469         A
## 5              655              473         A
## 6              660              478         A

```

Now train the model with 5 fold cross validation

```

fitControl <- trainControl(method = "cv",
                           number = 5,
                           allowParallel = TRUE)
fit <- caret::train(classe~.-user_name, method="rf", data=training, trControl = fitControl)
beep()

```

And check the results:

```

print(fit)

## Random Forest
##
## 19215 samples
##    53 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 15373, 15373, 15371, 15371, 15372
## Resampling results across tuning parameters:
##
##  mtry  Accuracy   Kappa
##    2    0.9940152 0.9924284
##   27    0.9941191 0.9925600
##   52    0.9889672 0.9860421
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.

print(fit$resample)

##      Accuracy      Kappa Resample
## 1 0.9927121 0.9907794   Fold1
## 2 0.9927159 0.9907840   Fold3
## 3 0.9945341 0.9930851   Fold2
## 4 0.9950559 0.9937456   Fold5
## 5 0.9955775 0.9944056   Fold4

print(confusionMatrix.train(fit))

## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##           A 28.4  0.1  0.0  0.0  0.0
##           B  0.0 19.2  0.1  0.0  0.0
##           C  0.0  0.0 17.3  0.2  0.0
##           D  0.0  0.0  0.1 16.2  0.0
##           E  0.0  0.0  0.0  0.0 18.3
##
## Accuracy (average) : 0.9941

```

The results are good, showing that results can be predicted with over 99% accuracy

Applying the resultin model to the test set

Finally we make predictions to the test set. For this case, we retrain the model over the entire training set to get a slightly higher reliability. We train the model and check the results

```

fitControl <- trainControl(method = "boot",
                           number = 1,

```

```

allowParallel = TRUE)

fitFinal <- train(classe~., method="rf",data=training,trControl = fitControl)
fitFinal

## Random Forest
##
## 19215 samples
##    53 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (1 reps)
## Summary of sample sizes: 19215
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9938668  0.9922353
##   29    0.9920126  0.9898890
##   57    0.9833119  0.9788757
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.

Predict on the test set:

testpredict<-predict(fitFinal,newdata = testing)
testpredict

## [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

Finally close the multithreading

stopCluster(cluster)
registerDoSEQ()

```