Assignment: Prediction Assignment Writeup

Antonio Maurandi López

21 de junio de 2016

Using devices such as *Jawbone Up*, *Nike FuelBand*, and *Fitbit* it is now possible to collect a large amount of data about personal activity relatively inexpensively. Our goal is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. We will use data data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the "classe" variable.

Loading and processing the data

Exploration of the data.

```
# download data from the source url
filepath     <- "XXXXX"
filepath     <- "pml-training.csv"
train <- read.table(filepath,  sep="," , header=T, dec=".")
filepath     <- "XXX"
filepath     <- "pml-testing.csv"
test <- read.table(filepath,  sep="," , header=T, dec=".")
```

Missing values: `NA`

There are variables with a lot of `NA` values, we will use only those variables which are not `NA` always, let set the criteria at 80% of data available, not `NA`.

```
n          <- nrow(train)
f.numofna <- function (vector){
    return( sum(is.na (vector)))
}

x       <- as.vector(apply(train, 2, f.numofna)/n)
pander(table(x), caption = "Number of variables with a percentaje of missiing values: `0%` or `97%`")
```

Table 1: Number of variables with a percentaje of missiing values: `0%` or `97%`

0	0.979308938946081
93	67

```
varsnona <- (x<0.8)

n1    <- length(names(train) )
train <- train[ ,varsnona ]
test  <- test[  ,varsnona ]
```

```r
# kk<-lapply(train,data.class)
# pander(kk)
n2 <- length( names(train) )
```

We will keep 93 variables instead of the original 160 variables.

Outcome variable `classe`

It is interesting to check if there are any class more presnet tahn others

```r
tt<-table(train$classe)
pander::pander( tt
                ,caption="absolute frecuency of classes of variebl classe" )
```

Table 2: absolute frecuency of classes of variebl classe

A	B	C	D	E
5580	3797	3422	3216	3607

```r
tt<- prop.table(tt)*100
pander::pander(tt
               , caption="percentage table" )
```

Table 3: percentage table

A	B	C	D	E
28.44	19.35	17.44	16.39	18.38

We can see that there are 5 different clasifications and that all of them are arround 16% and 29% of the total data available.

```r
pander(numSummary( train )[ ,-c(7:17)]
       , caption="Preliminary descriptives of numerical variables in the trainning dataset.")
```

Table 4: Preliminary descriptives of numerical variables in the trainning dataset.

	n	mean	sd	median	max	min
X	19622	9812	5665	9812	19622	1
raw_timestamp_part_1	19622	1322827119	204928	1322832920	1323095081	1322489605
raw_timestamp_part_2	19622	500656	288223	496380	998801	294
num_window	19622	430.6	247.9	424	864	1
roll_belt	19622	64.41	62.75	113	162	-28.9
pitch_belt	19622	0.3053	22.35	5.28	60.3	-55.8
yaw_belt	19622	-11.21	95.19	-13	179	-180

	n	mean	sd	median	max	min
total_accel_belt	19622	11.31	7.742	17	29	0
gyros_belt_x	19622	-0.005592	0.2073	0.03	2.22	-1.04
gyros_belt_y	19622	0.03959	0.07824	0.02	0.64	-0.64
gyros_belt_z	19622	-0.1305	0.2413	-0.1	1.62	-1.46
accel_belt_x	19622	-5.595	29.64	-15	85	-120
accel_belt_y	19622	30.15	28.58	35	164	-69
accel_belt_z	19622	-72.59	100.4	-152	105	-275
magnet_belt_x	19622	55.6	64.18	35	485	-52
magnet_belt_y	19622	593.7	35.68	601	673	354
magnet_belt_z	19622	-345.5	65.21	-320	293	-623
roll_arm	19622	17.83	72.74	0	180	-180
pitch_arm	19622	-4.612	30.68	0	88.5	-88.8
yaw_arm	19622	-0.6188	71.36	0	180	-180
total_accel_arm	19622	25.51	10.52	27	66	1
gyros_arm_x	19622	0.04277	1.994	0.08	4.87	-6.37
gyros_arm_y	19622	-0.2571	0.8514	-0.24	2.84	-3.44
gyros_arm_z	19622	0.2695	0.5532	0.23	3.02	-2.33
accel_arm_x	19622	-60.24	182	-44	437	-404
accel_arm_y	19622	32.6	109.9	14	308	-318
accel_arm_z	19622	-71.25	134.7	-47	292	-636
magnet_arm_x	19622	191.7	443.6	289	782	-584
magnet_arm_y	19622	156.6	201.9	202	583	-392
magnet_arm_z	19622	306.5	326.6	444	694	-597
roll_dumbbell	19622	23.84	69.93	48.17	153.5	-153.7
pitch_dumbbell	19622	-10.78	36.99	-20.96	149.4	-149.6
yaw_dumbbell	19622	1.674	82.52	-3.324	155	-150.9
total_accel_dumbbell	19622	13.72	10.23	10	58	0
gyros_dumbbell_x	19622	0.1611	1.509	0.13	2.22	-204
gyros_dumbbell_y	19622	0.04606	0.61	0.03	52	-2.1
gyros_dumbbell_z	19622	-0.129	2.287	-0.13	317	-2.38
accel_dumbbell_x	19622	-28.62	67.32	-8	235	-419
accel_dumbbell_y	19622	52.63	80.75	41.5	315	-189
accel_dumbbell_z	19622	-38.32	109.5	-1	318	-334
magnet_dumbbell_x	19622	-328.5	339.7	-479	592	-643
magnet_dumbbell_y	19622	221	326.9	311	633	-3600
magnet_dumbbell_z	19622	46.05	140	13	452	-262
roll_forearm	19622	33.83	108	21.7	180	-180
pitch_forearm	19622	10.71	28.15	9.24	89.8	-72.5
yaw_forearm	19622	19.21	103.2	0	180	-180
total_accel_forearm	19622	34.72	10.06	36	108	0
gyros_forearm_x	19622	0.158	0.6486	0.05	3.97	-22
gyros_forearm_y	19622	0.07517	3.101	0.03	311	-7.02
gyros_forearm_z	19622	0.1512	1.754	0.08	231	-8.09
accel_forearm_x	19622	-61.65	180.6	-57	477	-498
accel_forearm_y	19622	163.7	200.1	201	923	-632
accel_forearm_z	19622	-55.29	138.4	-39	291	-446
magnet_forearm_x	19622	-312.6	347	-378	672	-1280
magnet_forearm_y	19622	380.1	509.4	591	1480	-896
magnet_forearm_z	19622	393.6	369.3	511	1090	-973

```r
pander( charSummary(train)
        , caption="Preliminary descriptives of non numerical variables in the trainning dataset.")
```

Table 5: Preliminary descriptives of non numerical variables in the trainning dataset.

	n	miss	miss%	unique
user_name	19622	0	0	6
cvtd_timestamp	19622	0	0	20
new_window	19622	0	0	2
kurtosis_roll_belt	19622	0	0	397
kurtosis_picth_belt	19622	0	0	317
kurtosis_yaw_belt	19622	0	0	2
skewness_roll_belt	19622	0	0	395
skewness_roll_belt.1	19622	0	0	338
skewness_yaw_belt	19622	0	0	2
max_yaw_belt	19622	0	0	68
min_yaw_belt	19622	0	0	68
amplitude_yaw_belt	19622	0	0	4
kurtosis_roll_arm	19622	0	0	330
kurtosis_picth_arm	19622	0	0	328
kurtosis_yaw_arm	19622	0	0	395
skewness_roll_arm	19622	0	0	331
skewness_pitch_arm	19622	0	0	328
skewness_yaw_arm	19622	0	0	395
kurtosis_roll_dumbbell	19622	0	0	398
kurtosis_picth_dumbbell	19622	0	0	401
kurtosis_yaw_dumbbell	19622	0	0	2
skewness_roll_dumbbell	19622	0	0	401
skewness_pitch_dumbbell	19622	0	0	402
skewness_yaw_dumbbell	19622	0	0	2
max_yaw_dumbbell	19622	0	0	73
min_yaw_dumbbell	19622	0	0	73
amplitude_yaw_dumbbell	19622	0	0	3
kurtosis_roll_forearm	19622	0	0	322
kurtosis_picth_forearm	19622	0	0	323
kurtosis_yaw_forearm	19622	0	0	2
skewness_roll_forearm	19622	0	0	323
skewness_pitch_forearm	19622	0	0	319
skewness_yaw_forearm	19622	0	0	2
max_yaw_forearm	19622	0	0	45
min_yaw_forearm	19622	0	0	45
amplitude_yaw_forearm	19622	0	0	3
classe	19622	0	0	5

Standaritation

It's interestilng to *scale* and *center* the data it to get models less influenced by the diferent scale of the predictor variables.

```r
procValues <- preProcess( train, method = c("center", "scale") )
trainN     <- predict( procValues, train )
testN      <- predict( procValues, test )
```

We will work with the data set witout the classification variable `classe`.

```r
trainN <- trainN[ ,-c(1:2) ] # delete the id var and username
testN  <- testN[  ,-c(1:2) ]


myclasse     <- trainN$classe  # classification/outcome variable
trainN       <- subset(trainN, select = -c(classe) )
testN        <- subset(testN , select = -c(problem_id) )
```

There are varibles thar are not of the same type (class) in both datasets

```r
# problema con las classes de las variables aml! 20160621
k1 <- sapply( trainN, class)
k2 <- sapply( testN, class )
kk <- data.frame( k1, k2, k0 = NA , stringsAsFactors = FALSE )
# str(kk)
# head(kk)
for (i in 1: nrow(kk)){
    if (kk$k1[i]==kk$k2[i]) { kk$k0[i] <- TRUE }
    else{ kk$k0[i]<- FALSE}
}
length(rownames(kk))
```

[1] 90

```r
# rownames(kk)[kk$k0]
# there are variaboles thar are not of the same type (class) in both datasets
pander(kk[kk$k0==FALSE,1:2], caption="Variables in datasets thar are not of the same data class.")
```

Table 6: Variables in datasets thar are not of the same data class.

	k1	k2
kurtosis_roll_belt	factor	logical
kurtosis_picth_belt	factor	logical
kurtosis_yaw_belt	factor	logical
skewness_roll_belt	factor	logical
skewness_roll_belt.1	factor	logical
skewness_yaw_belt	factor	logical
max_yaw_belt	factor	logical
min_yaw_belt	factor	logical
amplitude_yaw_belt	factor	logical
kurtosis_roll_arm	factor	logical
kurtosis_picth_arm	factor	logical
kurtosis_yaw_arm	factor	logical
skewness_roll_arm	factor	logical
skewness_pitch_arm	factor	logical

	k1	k2
skewness_yaw_arm	factor	logical
kurtosis_roll_dumbbell	factor	logical
kurtosis_picth_dumbbell	factor	logical
kurtosis_yaw_dumbbell	factor	logical
skewness_roll_dumbbell	factor	logical
skewness_pitch_dumbbell	factor	logical
skewness_yaw_dumbbell	factor	logical
max_yaw_dumbbell	factor	logical
min_yaw_dumbbell	factor	logical
amplitude_yaw_dumbbell	factor	logical
kurtosis_roll_forearm	factor	logical
kurtosis_picth_forearm	factor	logical
kurtosis_yaw_forearm	factor	logical
skewness_roll_forearm	factor	logical
skewness_pitch_forearm	factor	logical
skewness_yaw_forearm	factor	logical
max_yaw_forearm	factor	logical
min_yaw_forearm	factor	logical
amplitude_yaw_forearm	factor	logical

```
vasrthatareidenticalinclass <- rownames(kk)[kk$k0]  # in boyh dataframes
trainN <- trainN [ ,kk$k0]
trainN       <- subset(trainN, select = vasrthatareidenticalinclass)
testN        <- subset(testN , select = vasrthatareidenticalinclass)
```

We will work only with those variables that have the same class in both datasets: raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window, num_window, roll_belt, pitch_belt, yaw_belt, total_accel_belt, gyros_belt_x, gyros_belt_y, gyros_belt_z, accel_belt_x, accel_belt_y, accel_belt_z, magnet_belt_x, magnet_belt_y, magnet_belt_z, roll_arm, pitch_arm, yaw_arm, total_accel_arm, gyros_arm_x, gyros_arm_y, gyros_arm_z, accel_arm_x, accel_arm_y, accel_arm_z, magnet_arm_x, magnet_arm_y, magnet_arm_z, roll_dumbbell, pitch_dumbbell, yaw_dumbbell, total_accel_dumbbell, gyros_dumbbell_x, gyros_dumbbell_y, gyros_dumbbell_z, accel_dumbbell_x, accel_dumbbell_y, accel_dumbbell_z, magnet_dumbbell_x, magnet_dumbbell_y, magnet_dumbbell_z, roll_forearm, pitch_forearm, yaw_forearm, total_accel_forearm, gyros_forearm_x, gyros_forearm_y, gyros_forearm_z, accel_forearm_x, accel_forearm_y, accel_forearm_z, magnet_forearm_x, magnet_forearm_y, magnet_forearm_z.

Still there are two variables with typoe `factor`: cvtd_timestamp, new_window. We will delete them because diferences in number of levels and `NA` valyues are usually a problem for random forest.

```
# qitamos los factores
names(trainN[,sapply(trainN,is.factor)])
```

[1] "cvtd_timestamp" "new_window"

```
names(testN[,sapply(testN,is.factor)])
```

[1] "cvtd_timestamp" "new_window"

```r
# trainN <- trainN[,-c(sapply(trainN,is.factor))]
# testN <- testN[,-c(sapply(testN,is.factor))]

 trainN <- trainN[ ,-c(3:4) ]
 testN  <- testN[  ,-c(3:4) ]
```

Now we have a trainning data set with 55 variables and 19622 observations.

- **Variables in the model**: *raw_timestamp_part_1, raw_timestamp_part_2, num_window, roll_belt, pitch_belt, yaw_belt, total_accel_belt, gyros_belt_x, gyros_belt_y, gyros_belt_z, accel_belt_x, accel_belt_y, accel_belt_z, magnet_belt_x, magnet_belt_y, magnet_belt_z, roll_arm, pitch_arm, yaw_arm, total_accel_arm, gyros_arm_x, gyros_arm_y, gyros_arm_z, accel_arm_x, accel_arm_y, accel_arm_z, magnet_arm_x, magnet_arm_y, magnet_arm_z, roll_dumbbell, pitch_dumbbell, yaw_dumbbell, total_accel_dumbbell, gyros_dumbbell_x, gyros_dumbbell_y, gyros_dumbbell_z, accel_dumbbell_x, accel_dumbbell_y, accel_dumbbell_z, magnet_dumbbell_x, magnet_dumbbell_y, magnet_dumbbell_z, roll_forearm, pitch_forearm, yaw_forearm, total_accel_forearm, gyros_forearm_x, gyros_forearm_y, gyros_forearm_z, accel_forearm_x, accel_forearm_y, accel_forearm_z, magnet_forearm_x, magnet_forearm_y, magnet_forearm_z.*

Fit a Model

Cross validation

We will use a 60% training set, 40% prove set of the total data set with clsification (`classe`).

```r
set.seed( pi )
casostest1  <- createDataPartition( myclasse, p=0.6, list = FALSE )

train1      <- trainN [ casostest1, ]  # trainning data set
train2      <- trainN [-casostest1, ] # proving data set
```

We will use to build a model a trainning data set of 11776 observations that is the 60% of the data in the origibal trainning dataset. We will test the model in a test data set of 7846 that is the remaining 40% of the data of the originla trainning dataset.

We will use the function `randomForest()` from `randomForest` package to fit the model.

```r
# library( "randomForest" )
#  This function can not work with `factor` variables thar have more than 54 levels
# , so we will limitate the factor to have no mor than 10 levels
# , using an _ad hoc_ function `flevels()`.
# so we create a function to select only those fcator variables
# with less than a certain numeber (nl) of levels
# mynl <- 9
# flevels <- function(v, nl = mynl ){
#     if (nlevels(v)< nl) return (TRUE)
#     else return(FALSE)
# }
system.time(
     rfo2 <- randomForest( myclasse[casostest1] ~. , data = train1
    # rfo2 <- randomForest( myclasse[casostest1] ~. , data = train1[,sapply(train1, flevels)]
```

```
                    # , mtry = 7  # el default es raiz(p)/3, donde p es el num de vars
                    # , subset = train
                    , importance = TRUE )
    )
```

user system elapsed 56.312 0.072 56.392

```
# varsinmodel <- names( train1[,sapply(train1, flevels)] )
```

```
pander( importance( rfo2 ) )
```

Table 7: Table continues below

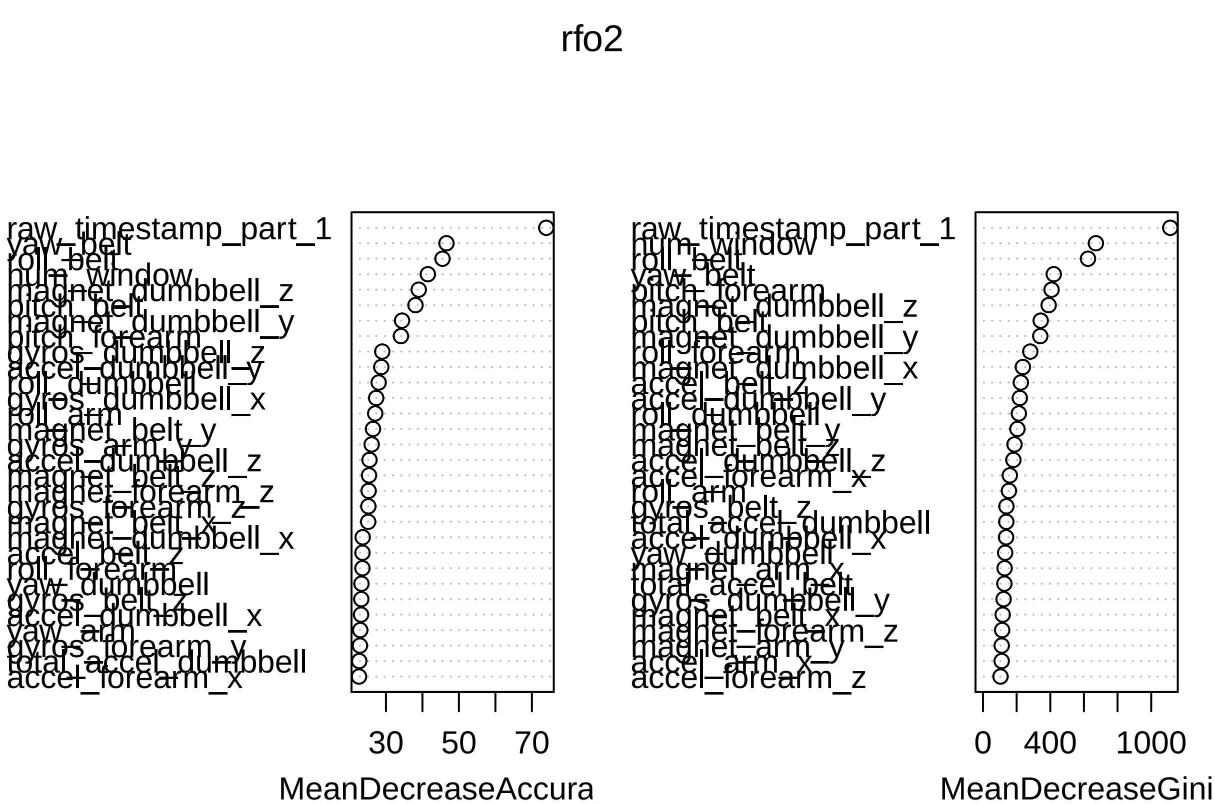
	A	B	C	D	E	MeanDecreaseAccuracy
raw_timestamp_part_1	50.48	55.66	58.21	61.13	40.81	73.94
raw_timestamp_part_2	9.755	9.758	9.134	7.915	7.731	18.4
num_window	28.95	37.23	44.09	36.21	34.17	41.49
roll_belt	31.47	40.42	35.44	39.81	35.37	45.5
pitch_belt	24.54	35.23	30.03	28.86	27.02	38.1
yaw_belt	31.23	33.97	31.8	38.06	27.01	46.59
total_accel_belt	13.58	14.26	12.43	13.78	15.22	16.07
gyros_belt_x	12.62	13.8	15.8	11.04	13.45	19.44
gyros_belt_y	9.607	12.96	13.58	12.83	14.38	16.72
gyros_belt_z	16.09	21.29	20.3	18.56	20.21	23.24
accel_belt_x	12.73	16.34	15.72	13.81	13.29	18.67
accel_belt_y	10.36	12.05	11.51	14.22	12.34	14.24
accel_belt_z	17.06	22.3	20.34	19.52	19.06	23.57
magnet_belt_x	13.7	21.26	20.4	17.45	20.19	25.13
magnet_belt_y	19.51	22.94	23.09	23.89	21.3	26.45
magnet_belt_z	18.5	21.85	20.94	24.89	20.19	25.36
roll_arm	16.68	24.61	22.39	22.87	19.51	27.06
pitch_arm	12.33	20.63	15.44	15.21	14.11	20.01
yaw_arm	16.57	19.56	19.17	20.58	16.94	22.96
total_accel_arm	8.42	19.12	15.89	16.01	13.71	21.34
gyros_arm_x	12.28	17.29	12.6	16.71	15.25	18.34
gyros_arm_y	11.74	21.88	18.37	19.48	16.94	26.09
gyros_arm_z	10.37	14.07	10.28	13.16	11.85	21.75
accel_arm_x	12.8	14.75	14.18	17.05	12.82	15.55
accel_arm_y	13.38	17.02	14.1	16.34	14.81	21.52
accel_arm_z	10.01	14.16	15.62	15.5	11.49	15.57
magnet_arm_x	15.43	15.04	17.16	16.44	13.97	16.99
magnet_arm_y	12.15	15.57	17.02	17.85	12.88	16.23
magnet_arm_z	15.05	21.12	17.45	16.46	15.97	22.2
roll_dumbbell	20.53	24.48	25.24	25.36	23.37	28.01
pitch_dumbbell	10.5	17.94	14.81	12.8	13.38	15.73
yaw_dumbbell	15.79	19.34	20.46	19.18	20.52	23.29
total_accel_dumbbell	15.4	19.76	18.74	20.25	18.83	22.69
gyros_dumbbell_x	13.49	18.46	17.46	15.41	15.47	27.32
gyros_dumbbell_y	15.81	17.45	21.15	17.16	15.61	19.66
gyros_dumbbell_z	13.86	19.67	12.98	14.62	11.09	28.99
accel_dumbbell_x	16.23	22.24	18.73	17.95	19.38	23.18
accel_dumbbell_y	21.78	23.24	25.88	23.81	22.99	28.72

	A	B	C	D	E	MeanDecreaseAccuracy
accel_dumbbell_z	16.47	21.62	22.61	21.46	22.58	25.49
magnet_dumbbell_x	20.08	21.86	23.87	22.71	20.67	23.63
magnet_dumbbell_y	25.53	31.98	35.21	30.1	25.82	34.41
magnet_dumbbell_z	36.55	30.84	36.36	31.24	30.31	38.98
roll_forearm	23.42	20.69	25.51	18.86	19.81	23.56
pitch_forearm	25.62	28.5	32.69	29.35	27.43	34.11
yaw_forearm	13.54	16.25	15.47	16.59	15.03	19.98
total_accel_forearm	14.21	15.37	16.01	12.72	15.43	19.9
gyros_forearm_x	9.025	13.86	14.26	12.9	13	22.44
gyros_forearm_y	11.22	19.28	17.99	15.9	15.58	22.87
gyros_forearm_z	9.701	18.26	16.87	13.43	12.07	25.18
accel_forearm_x	15.38	21.28	19.62	24.03	20.21	22.62
accel_forearm_y	14.46	17.18	17.39	15.92	17.22	21.3
accel_forearm_z	12.69	15.43	18.48	16.39	15.78	19.2
magnet_forearm_x	12.74	17.22	16.22	15.44	17.08	18.84
magnet_forearm_y	14.53	17.17	17.02	16.26	16.33	19.67
magnet_forearm_z	16.72	21.06	18.98	20.4	19.46	25.25

	MeanDecreaseGini
raw_timestamp_part_1	1113
raw_timestamp_part_2	13.11
num_window	671.1
roll_belt	624.1
pitch_belt	343
yaw_belt	420.7
total_accel_belt	127.1
gyros_belt_x	43.5
gyros_belt_y	53.34
gyros_belt_z	139.2
accel_belt_x	64.63
accel_belt_y	74.77
accel_belt_z	224.8
magnet_belt_x	117.1
magnet_belt_y	204.5
magnet_belt_z	187.2
roll_arm	154
pitch_arm	73.85
yaw_arm	103.4
total_accel_arm	42.77
gyros_arm_x	51.75
gyros_arm_y	51.18
gyros_arm_z	25.59
accel_arm_x	111
accel_arm_y	67.32
accel_arm_z	57.67
magnet_arm_x	128.3
magnet_arm_y	111.3
magnet_arm_z	74.52
roll_dumbbell	213.1
pitch_dumbbell	96.71
yaw_dumbbell	131.7

	MeanDecreaseGini
total_accel_dumbbell	138.5
gyros_dumbbell_x	50.82
gyros_dumbbell_y	122.5
gyros_dumbbell_z	33.97
accel_dumbbell_x	137.2
accel_dumbbell_y	218.8
accel_dumbbell_z	180.5
magnet_dumbbell_x	236.9
magnet_dumbbell_y	340.4
magnet_dumbbell_z	390.3
roll_forearm	281.1
pitch_forearm	407.6
yaw_forearm	68.97
total_accel_forearm	45.39
gyros_forearm_x	32.97
gyros_forearm_y	48.6
gyros_forearm_z	31.7
accel_forearm_x	159.4
accel_forearm_y	60.36
accel_forearm_z	104
magnet_forearm_x	96.95
magnet_forearm_y	91.98
magnet_forearm_z	113.9

```
varImpPlot( rfo2 )
```



Testing the model with `train2` (40% of the trainning data)

Now we will test or model with the trainning data set that reprsenetn de 40% of the original trainning data set: `train2`.

```
train2_prediction <- predict( rfo2, newdata = train2 )
kk                <- confusionMatrix( train2_prediction, myclasse[-casostest1] )
pander(kk$overall[1:2], caption = "overall." )
```

Accuracy	Kappa
0.9978	0.9973

```
pander(kk$table       , caption = "Confusion table." )
```

Table 10: Confusion table.

	A	B	C	D	E
A	2232	7	0	0	0
B	0	1511	2	0	0
C	0	0	1366	1	0
D	0	0	0	1285	7
E	0	0	0	0	1435

```
pander(t(kk$byClass)  , caption = "model parameters by class." )
```

Table 11: model parameters by class.

	Class: A	Class: B	Class: C	Class: D	Class: E
Sensitivity	1	0.9954	0.9985	0.9992	0.9951
Specificity	0.9988	0.9997	0.9998	0.9989	1
Pos Pred Value	0.9969	0.9987	0.9993	0.9946	1
Neg Pred Value	1	0.9989	0.9997	0.9998	0.9989
Prevalence	0.2845	0.1935	0.1744	0.1639	0.1838
Detection Rate	0.2845	0.1926	0.1741	0.1638	0.1829
Detection Prevalence	0.2854	0.1928	0.1742	0.1647	0.1829
Balanced Accuracy	0.9994	0.9975	0.9992	0.9991	0.9976

```
# confusionMatrix(train2_prediction, myclasse[-casostest1])$overall['Accuracy']
```

We obtain a higth accuracy 0.998, so we may think that we have a goog model for prediction.

Clasification of new cases

```r
testN_prediction <- predict(rfo2, newdata=testN)
pander(data.frame(test$user_name,testN_prediction))
```

test.user_name	testN_prediction
pedro	B
jeremy	A
jeremy	B
adelmo	A
eurico	A
jeremy	E
jeremy	D
jeremy	B
carlitos	A
charles	A
carlitos	B
jeremy	C
eurico	B
jeremy	A
jeremy	E
eurico	E
pedro	A
carlitos	B
pedro	B
eurico	B

Sessioninfo()

```r
sessionInfo()
```

R version 3.2.2 (2015-08-14) Platform: i686-pc-linux-gnu (32-bit) Running under: Ubuntu 15.10

locale: [1] LC_CTYPE=es_ES.UTF-8 LC_NUMERIC=C
[3] LC_TIME=es_ES.UTF-8 LC_COLLATE=es_ES.UTF-8
[5] LC_MONETARY=es_ES.UTF-8 LC_MESSAGES=es_ES.UTF-8
[7] LC_PAPER=es_ES.UTF-8 LC_NAME=C
[9] LC_ADDRESS=C LC_TELEPHONE=C
[11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C

attached base packages: [1] stats graphics grDevices utils datasets methods base

other attached packages: [1] randomForest_4.6-12 xda_0.1 devtools_1.10.0
[4] caret_6.0-68 ggplot2_2.0.0 lattice_0.20-33
[7] pander_0.6.0

loaded via a namespace (and not attached): [1] codetools_0.2-14 digest_0.6.9 htmltools_0.3
[4] minqa_1.2.4 splines_3.2.2 MatrixModels_0.4-1 [7] scales_0.3.0 grid_3.2.2 stringr_1.0.0
[10] e1071_1.6-7 knitr_1.12.3 lme4_1.1-11
[13] munsell_0.4.3 nnet_7.3-10 foreach_1.4.3
[16] iterators_1.0.8 mgcv_1.8-7 Matrix_1.2-2
[19] MASS_7.3-43 plyr_1.8.3 stats4_3.2.2

[22] stringi_1.0-1 pbkrtest_0.4-5 magrittr_1.5
[25] car_2.1-1 reshape2_1.4.1 rmarkdown_0.9.2
[28] evaluate_0.8 gtable_0.1.2 colorspace_1.2-6
[31] yaml_2.1.13 tools_3.2.2 parallel_3.2.2
[34] nloptr_1.0.4 nlme_3.1-124 quantreg_5.19
[37] class_7.3-13 formatR_1.2.1 memoise_1.0.0
[40] Rcpp_0.12.3 SparseM_1.7