Prediction Assignment

Ricardo Gomes César

11/05/2020

Executive Summary

The goal in this work was to develop models to estimate the Classe parameter from the "Human Active Recognition" dataset, which collects vast data on human movement to infer on the quality of exercise, among other variables. Initially we refined the data, removing parameters with >10% blank values, autocorrelated values and near zero variances values. We also removed parameters related to the data provider ID, since these would not contribute to datasets other than the training. We developed a Random Forest model with the remaining parameters to estimate the Classe parameter on the training data, and estimated these values for the test data provided. Our results were uploaded on GitHub on the following link: http://graphics8.nytimes.com/images/2008/04/16/us/0416-nat-subOBAMA.jpg

Preparing the dataset

```
pml_training <- read.csv('pml-training.csv')
pml_testing <- read.csv('pml-testing.csv')
```

After loading the huge dataset provided. I removed collums that contained >10% of blanck (NA) values:

```
pml_training[pml_training==""]<-NA

pml_training <- pml_training[, colSums(is.na(pml_training)) < (nrow(pml_training)*0.1)]
```

After that, I looked for correlated variables

```
pml.cor <- cor(pml_training[,-which(sapply(pml_training, class) == "factor")])

#removing the lower layer
for (i in 2:56) {
  pml.cor[i,1:(i-1)] <- 0

}

#Keeping only those which correlation is <0.8 or >0.8
pml.cor.loc <- which(pml.cor > 0.8 | pml.cor < -0.8,arr.ind = TRUE )

pml.cor.loc <- as.data.frame(pml.cor.loc)
```

```
#Removing variables that are correlated with themselves (row = collum
pml.cor.loc1 <- data.frame(row=as.integer(), col=as.integer())

for(i in 1:nrow(pml.cor.loc)){
if (pml.cor.loc$row[i] != pml.cor.loc$col[i]){
  pml.cor.loc1[i,] <- pml.cor.loc[i,]
} else {
  pml.cor.loc1[i,] <- c('NA', 'NA')

}
}

#removing rows that are NA
pml.cor.loc1 <- as.data.frame(pml.cor.loc1)

pml.cor.loc2 <- pml.cor.loc1[pml.cor.loc1 != 'NA', ]
pml.cor.loc2 <- na.omit(pml.cor.loc2)

pml.cor.loc2$row <- as.integer(pml.cor.loc2$row)
pml.cor.loc2$col <- as.integer(pml.cor.loc2$col)

#Finding the correlations

correlations <- data.frame()

for (i in 1:nrow(pml.cor.loc2)) {
  correlations[i,1] <- rownames(pml.cor)[pml.cor.loc2$row[i]]
  correlations[i,2] <- colnames(pml.cor)[pml.cor.loc2$col[i]]
}


names.corr <- unique(correlations)
temp1 <- unique(names.corr$V1)
temp2 <- unique(names.corr$V2)

names.corr <- unique(correlations)

library(dplyr)
pml_training1 <- select(pml_training, -c(temp1, temp2))
```

This made us remove the following parameters (additional to those removed due to NAs)

```
                V1              V2
1         roll_belt         yaw_belt
2         roll_belt total_accel_belt
3        pitch_belt     accel_belt_x
4         roll_belt     accel_belt_y
5  total_accel_belt     accel_belt_y
6         roll_belt     accel_belt_z
7  total_accel_belt     accel_belt_z
8      accel_belt_y     accel_belt_z
9        pitch_belt    magnet_belt_x
10     accel_belt_x    magnet_belt_x
```

```
11      gyros_arm_x      gyros_arm_y
12      accel_arm_x     magnet_arm_x
13     magnet_arm_y     magnet_arm_z
14 gyros_dumbbell_x gyros_dumbbell_z
15   pitch_dumbbell accel_dumbbell_x
16     yaw_dumbbell accel_dumbbell_z
17 gyros_dumbbell_x  gyros_forearm_z
18 gyros_dumbbell_z  gyros_forearm_z
19  gyros_forearm_y  gyros_forearm_z
```

I removed the following variables related to identification of samples, since they will not be useful for testing data

```
[1] "X"                    "user_name"            "raw_timestamp_part_1" "raw_timestamp_part_2"
[5] "cvtd_timestamp"       "new_window"           "num_window"
```

We looked for and removed parameters with near zero variance.

```
library('caret')
zero.var <- nearZeroVar(pml_training, saveMetrics = TRUE, names = TRUE)
```

However, there was no Near Zero Variance parameters in the dataset with the remnant parameters.

Generating the model to estimate Classe

We used Random Forets in the package 'ranger' to estimate the Classe values based on the remnant 30 parameters after data processing (of the original 159 parameters).

```
library('ranger')
modelfit <- train(classe ~ ., method = 'ranger', data = pml_training2)

trainingpredict <- predict(modelfit, pml_training2)

table(trainingpredict, pml_training2$classe)
```

Finally, we included our estimated values for "Classe" parameter in the testing dataset

```
pred_test <- predict(modelfit, pml_testing)
pml_testing$classe <- pred_test
pred_test
```

```
 [1] "B" "A" "B" "A" "A" "E" "D" "B" "A" "A" "B" "C" "B" "A" "E" "E" "A" "B" "B" "B"
```