Fit Classifier

Giuseppe Fumarola

12/11/2020

Fit classifier

Summary

Write summary here

Load and preprocess

Data and libraries are loaded. Then data are preprocessed. A few variables have been excluded, since their measurements is not related to the physical activity, but rather to the device. The timestamp is parsed as its relevant class. The output variable is factorised.

There are 67 variables containing missing values. Those variables are removed.

33 near zero variance variables are removed as well. They mainly belong to measures of skewness and kurtosis for other variables.

The dataset is then split in train/test parts, with proportions 0.75 / 0.25.

```r
library(ggplot2)
library(caret)
library(naniar)
library(rattle)
library(dplyr)


rdata <- read.csv("~/R/R-coursera/pml-training.csv", row.names=1)
rexercise <- read.csv("~/R/R-coursera/pml-testing.csv", row.names=1)

rdata <- select(rdata, -c(2,3,5,6))
rexercise <- select(rexercise, -c(2,3,5,6))

rdata$cvtd_timestamp <- as.Date.character(rdata$cvtd_timestamp)

rexercise$cvtd_timestamp <- as.Date.character(rexercise$cvtd_timestamp)
rdata$classe <- as.factor(rdata$classe)

misstable <- miss_var_summary(rdata)
missvar <- subset(misstable, n_miss>0, variable)
rdata <- select(rdata, - (missvar)[[1]])
```

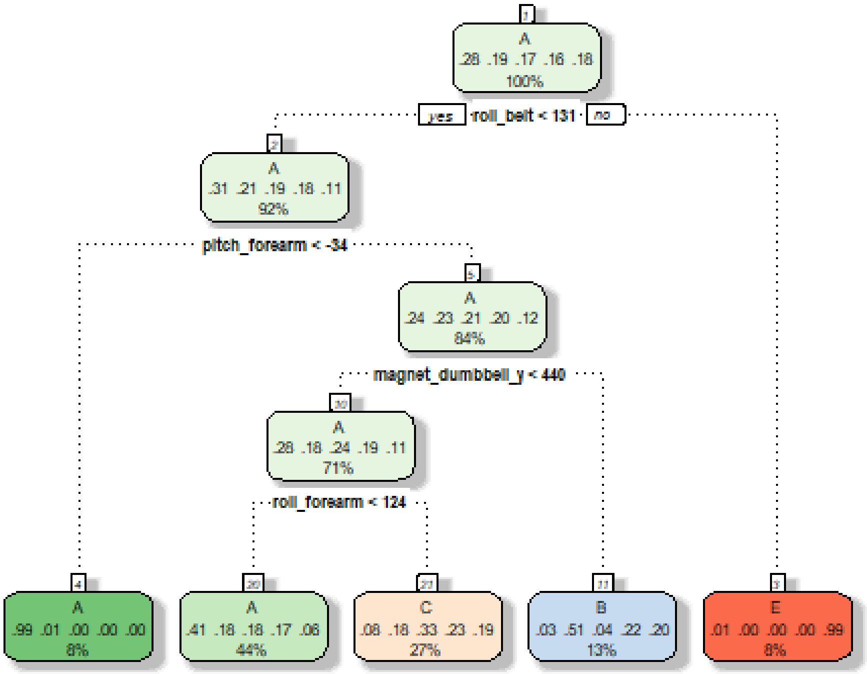
```
nzv <- nearZeroVar(rdata, saveMetrics= TRUE)
nzvar <- row.names(filter(nzv, nzv == TRUE))
pretrain <- select(rdata, - nzvar)

inTrain = createDataPartition(rdata$classe, p = 3/4)[[1]]
train = pretrain[inTrain,]
test = pretrain[-inTrain,]
```

Predictive model

At first, a tree decisional model is fit to predict the outcome variable. The output is shown in the chart below, which also displays the main predictive factors.

```
rp <- train(classe ~ ., data = train, method = "rpart")
fancyRpartPlot(rp$finalModel)
```



Rattle 2020-nov-13 00:13:08 User

However, the accuracy of the model is quite poor, around 50%. Despite being much better than random guessing, which holds a predictive power of 20%, a more sophisticated random forest model is fit, to improve the results. Nonetheless, the improvement is not very significant.

```
rf <- train(classe ~ ., data = train, method = "rf", ntree = 100, maxnodes = 10)
confusionMatrix(rf)
```

```
## Bootstrapped (25 reps) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 25.2  7.4  7.3  6.8  2.4
##          B  0.5  6.3  0.5  1.0  2.1
##          C  2.6  5.0  9.7  6.4  5.1
##          D  0.0  0.4  0.0  1.9  0.3
##          E  0.0  0.3  0.1  0.3  8.4
##
##  Accuracy (average) : 0.515
```

Testing the model

The model is finally controlled on the test dataset.

```
prediction <- predict(rf, test)
table(prediction, test$classe)
```

```
##
## prediction    A    B    C    D    E
##          A 1278  394  389  335  137
##          B   19  290   33    9   89
##          C   98  242  433  321  232
##          D    0   23    0  139   17
##          E    0    0    0    0  426
```

```
accuracy <- sum(diag(table(prediction, test$classe))) / sum(table(prediction,
test$classe))
accuracy
```

```
## [1] 0.5232463
```

The final model appears to have an out of sample error of almost 52% and its predictive power is much stronger for events of class A.