Practical Machine Learning Course Project

Pablo Portillo Garrigues

30 de mayo de 2018

Summary

The porpouse of this paper is to decide, using machine learning techniques, in what way was carried out the excercise. The outcome variable (Classe) is a factor variable with 5 leves. We pretend to use the data collected by different weareables in order to train 3 models. These models were trained using 5-fold Cross-Validation and we have tested the trained models with 20% if the data collected as testing sample. The 3 prediction functions used were: Random Forest, a boosted predictor and a linear discriminant analysis.

Loading data

First of all, we are going to load the datasets, we have indicated the different possibles outliers (NAs). We have also randomly sorted the data in order to break any possible in-row dependecy.

```
dat<-read.csv(file="pml-training.csv",sep=",",na.strings = c("NA",""))
datAux<-dat
set.seed(69)
dat<-dat[sample(nrow(dat)),]
validation<-read.csv(file="pml-testing.csv",sep=",",na.strings = c("NA",""))
```

NA Function with threshold

We have implemented a function wich returns the columns that have a higher ratio of outliers than the threshold indicated.

```
NAfunction<-function(df1,threshold){
    res<-c()
    for(i in 1:dim(df1)[2]){
        aux<-sum(is.na(df1[,i]))
        if((aux/dim(df1)[1])>=threshold){
            res<-c(res,i)
        }
    }
    res
}
```

Data Exploration

We have broke the dataset in to 2 samples, a training set (80% of the total set) and a test set.

```
inTrain <- createDataPartition(y=dat$classe, p=0.8, list=FALSE)
set.seed(323)
training<-dat[inTrain,]
testing<-dat[-inTrain,]
nzv<-nearZeroVar(training)
training<-training[,-1]
testing<-testing[,-1]
training<-training[,-nzv]
elNA<-NAfunction(training,0.5)
training<-training[,-elNA]
testing<-testing[,-nzv]
testing<-testing[,-elNA]
skimmed <- skim_to_wide(training)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
skimmed[, c(1:5, 9:11, 13, 15:16)]
```

```
## # A tibble: 55 x 11
##    type   variable  missing complete n     mean   sd     p0    p50   p100
##    <chr>  <chr>     <chr>   <chr>    <chr> <chr>  <chr>  <chr> <chr> <chr>
##  1 factor classe    0       15699    15699 <NA>   <NA>   <NA>  <NA>  <NA>
##  2 factor cvtd_tim~ 0       15699    15699 <NA>   <NA>   <NA>  <NA>  <NA>
##  3 factor new_wind~ 0       15699    15699 <NA>   <NA>   <NA>  <NA>  <NA>
##  4 factor user_name 0       15699    15699 <NA>   <NA>   <NA>  <NA>  <NA>
##  5 integ~ accel_ar~ 0       15699    15699 "  -5~ "   1~ " -4~ "  -~ "  4~
##  6 integ~ accel_ar~ 0       15699    15699 "   3~ "   1~ " -3~ "   ~ "  3~
##  7 integ~ accel_ar~ 0       15699    15699 "  -7~ "   1~ " -6~ "  -~ "  2~
##  8 integ~ accel_be~ 0       15699    15699 "   -~ "    ~ "  -~ "  -~ "   ~
##  9 integ~ accel_be~ 0       15699    15699 "   3~ "    ~ "  -~ "   ~ "  1~
## 10 integ~ accel_be~ 0       15699    15699 "  -7~ "   1~ " -2~ " -1~ "  1~
## # ... with 45 more rows, and 1 more variable: hist <chr>
```

We have eliminated the variables that had near to zero variability as well as the variables that had a ratio of outlier higher than 50%. We have eliminated these variables in the 3 datasets.

Training

We are first going to train a Random Forest algorithm with a 5-fold Cross-Validation

```
fitControl1 <- trainControl(method = "cv",
                           number = 5,
                           allowParallel = TRUE)

modRF1 <- train(classe~., method="rf",data=training,trControl = fitControl1,preProcess=c("center","scale"))
```

We are first going to train a boosted predictor algorithm with a 5-fold Cross-Validation

```
fitControlGBM <- trainControl(method = "cv",
                           number = 5,
                           allowParallel = TRUE)

modGBM1 <- train(classe~., method="gbm",data=training,trControl = fitControlGBM,preProcess=c("center","scale"))
```

We are first going to train a linear discriminant analysis algorithm with a 5-fold Cross-Validation

```
fitControlLDA <- trainControl(method = "cv",
                           number = 5,
                           allowParallel = TRUE)

modLDA1 <- train(classe~., method="lda",data=training,trControl = fitControlLDA,preProcess=c("center","scale"))
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear
```

testing

We are now going to test all the models with the testing part.

```
predRF<-predict(modRF1,newdata=testing)
predGBM<-predict(modGBM1,newdata=testing)
predLDA<-predict(modLDA1,newdata=testing)
confusionMatrix(predRF,testing$classe)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9992353      0.9990328      0.9977668      0.9998423      0.2844762
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

```
confusionMatrix(predGBM,testing$classe)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9974509      0.9967761      0.9953172      0.9987770      0.2844762
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

```
confusionMatrix(predLDA,testing$classe)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.8536834      0.8149392      0.8422334      0.8646038      0.2844762
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

As we can see, the three models seem to perform equally good.

Validation

We are going to test the three models with the validation data set that was provided

```
data.frame(RF=predict(modRF1,newdata=validation),GBM=predict(modGBM1,newdata=validation),LDA=predict(modLDA1,newd
ata=validation))
```

```
##    RF GBM LDA
## 1   B   B   B
## 2   A   A   B
## 3   B   B   B
## 4   A   A   A
## 5   A   A   A
## 6   E   E   E
## 7   D   D   D
## 8   B   B   C
## 9   A   A   A
## 10  A   A   A
## 11  B   B   B
## 12  C   C   C
## 13  B   B   B
## 14  A   A   A
## 15  E   E   E
## 16  E   E   E
## 17  A   A   A
## 18  B   B   B
## 19  B   B   B
## 20  B   B   B
```