PracticalMachineLearning

Emiliano Olmedo

23/1/2021

Summary Final Project: How well they do it.

Note: English is not my mother tongue, I apologize for any grammatical errors.

As the original paper states:

This human activity recognition research has traditionally focused on discriminating between different activities, i.e. to predict "which" activity was performed at a specific point in time... The "how (well)" investigation has only received little attention so far, even though it potentially provides useful information for a large variety of applications,such as sports training.

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

The original paper can be found: Read more: http:/groupware.les.inf.puc-rio.br/har#ixzz4TjqLnvVK

So our task will be build a predictive model using machine learning to classify correctly some unlabel cases using a know data of some people who took part in the experiment doing the exercises with different degrees of correction.

PreProcessing of the Data

We will be looking at the dimension at each step because not all the columns gives us information of interest, or rather useful to our analysis.

```
inTrain  <- createDataPartition(dat$classe, p=0.7, list=FALSE)
TrainSet <- dat[inTrain, ]
TestSet  <- dat[-inTrain, ]
dim(TrainSet)
```

```
## [1] 13737   160
```

```
dim(TestSet)
```

```
## [1] 5885  160
```

We drop those columns that had NA values, and we end with almost half the columns we begin with. Also we had columns that shouldn't be part of our analysis such as user_name etc.

```
TrainSet <- TrainSet[colSums(is.na(TrainSet)) == 0]
TestSet <- TestSet[colSums(is.na(TestSet)) == 0]
dim(TrainSet)
```

```
## [1] 13737    93
```

```
dim(TestSet)
```

```
## [1] 5885   93
```

```
unique(TrainSet$user_name)
```

```
## [1] "carlitos" "pedro"    "adelmo"   "charles"  "eurico"   "jeremy"
```

For that reason and with some exploratory analysis of the column names we choose to eliminate those columns with the following characteristics. As the outcome classe is a factor variable we make sure that R understands it as such.

```
#getting rid of more useless columns
TrainSet <- select(TrainSet, -contains("timestamp"), -ends_with("window"), -starts_with("user"), -X)
TestSet <- select(TestSet, -contains("timestamp"), -ends_with("window"), -starts_with("user"), -X)
dim(TrainSet)
```

```
## [1] 13737    86
```

```
dim(TestSet)
```

```
## [1] 5885   86
```

```
# "classe" variable to a factor variable
TrainSet$classe <- as.factor(TrainSet$classe)
TestSet$classe <- as.factor(TestSet$classe)
```

As per suggestion of the instructor it is important as part of our pre-processing analysis to search for useful variables such as those who doesn't have near zero variance.

```
temporal<-TrainSet$classe
temporaltest<-TestSet$classe
#making sure that the classe variable does not disappear due to this procedure.

nsv <- nearZeroVar(TrainSet)
TrainSet <- TrainSet[, -nsv]
TestSet  <- TestSet[, -nsv]
dim(TrainSet)
```

```
## [1] 13737    53
```

```
dim(TestSet)
```

```
## [1] 5885   53
```

Analysis of the Data

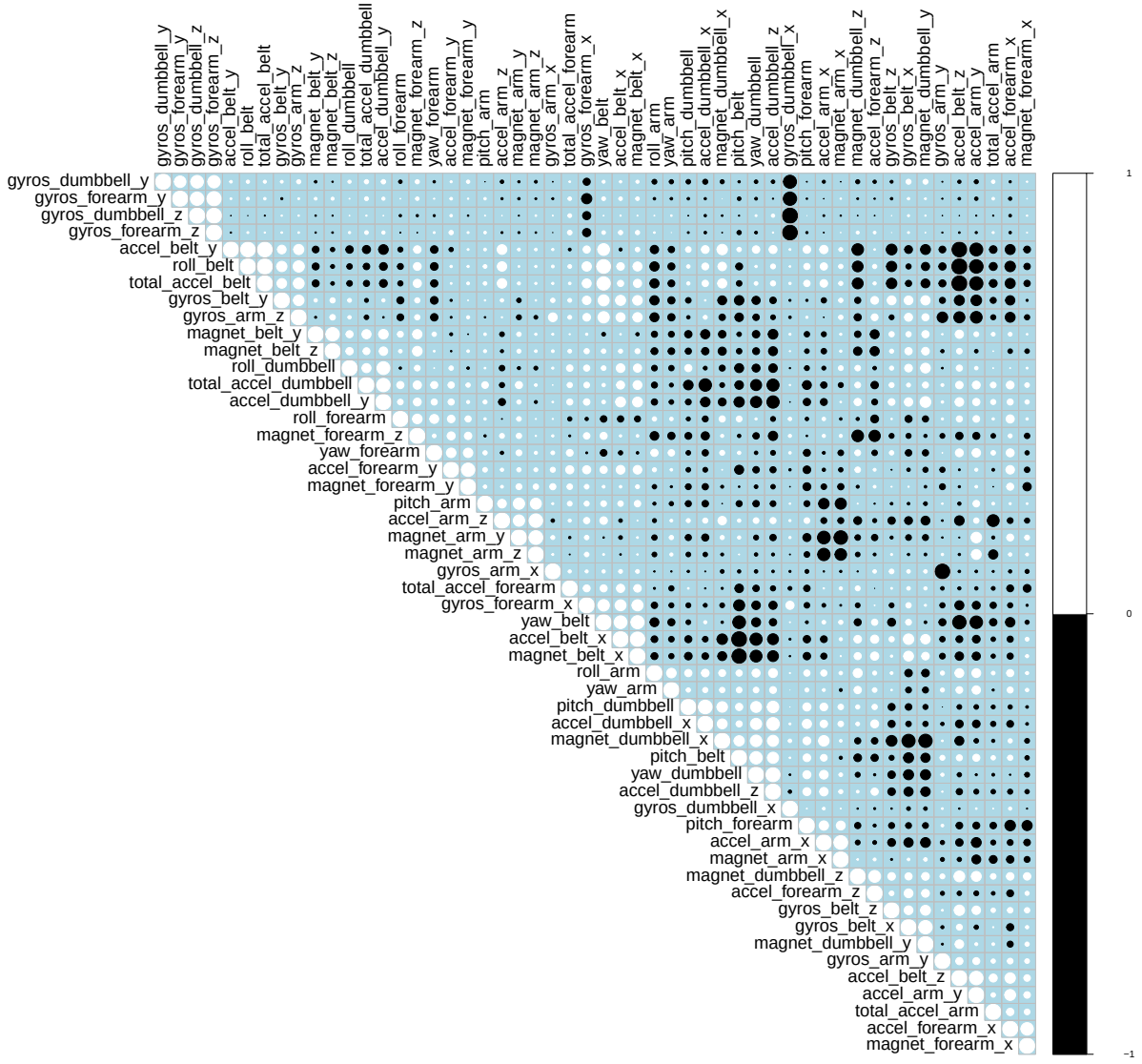
which(colnames(TrainSet)==‘classe’) shows the position of the column classe, we had to keep track of of its position.

We can see with the correlation plot, the different correlation between the variables but also that most of the names are referring to the different accelerations of the axes. This validates the selection of our variables for our analysis, if we are intended to classify different types of the quality of each exercise with the movement of the body the acceleration and its directions are fundamental.

```
which( colnames(TrainSet)=='classe' )
```

```
## [1] 53
```

```
corMatrix <- cor(TrainSet[, -53])
corrplot(corMatrix, order = "hclust",col = c("black", "white"),
         bg = "lightblue",type = 'upper',tl.cex = 1.5, tl.col = rgb(0, 0, 0))
```



We develop two types of analysis, the first will be done with naive bayes and the second with quadratic discriminant analysis, for naive bayes the next line is critical, if i use the default Train control features nb, at least with my computer the analysis is unfeasible (more than five minutes and still had not finished), so we need to override those and although it will take some time to finish is feasible.

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

The accuracy of this model its around 60%.

```
modelFit <- train(classe~.,preProcess='pca',data = TrainSet,method='nb',trControl = fitControl,tuneGrid
which( colnames(TestSet)=='classe' )
```

```
## [1] 53
```

```
prediction<-predict(modelFit, TestSet[,-53])
confusionMatrix(TestSet$classe,prediction)$overall['Accuracy']
```

```
##  Accuracy
## 0.6373832
```

We have to predict the answers quiz so with this analysis the predictions are the following.

```
predictionAnswers<-predict(modelFit,answersQuiz)
predictionAnswers
```

```
##  [1] C A A A A B D D A A A C B A E B A E E B
## Levels: A B C D E
```

this method gives us an accuracy around 75%

```
modelFit2 <- train(classe~.,preProcess='pca',data = TrainSet,method='qda' )
prediction2<-predict(modelFit2, TestSet[,-53])
confusionMatrix(TestSet$classe,prediction2)$overall['Accuracy']
```

```
##  Accuracy
## 0.7284622
```

```
predictionAnswers2<-predict(modelFit2,answersQuiz)
predictionAnswers2
```

```
##  [1] C A C A C B D B A A A C B A E E A B B B
## Levels: A B C D E
```

```
length(predictionAnswers2)
```

```
## [1] 20
```

```
length(predictionAnswers)
```

```
## [1] 20
```

final conclusions

Finally we have to make a decision about the classification, the best we can do from my point of view is to weigh the two methods, which will give us some security when answering the quiz.

Other methods can be used in the classification such as random forest or decision trees, if it is necessary to choose only one method I would take QDA for the obvious reason that it has greater accuracy.

A = 1 B = 2 C = 3 D = 4 E = 5

When the values coincide we have 1, otherwise 0

```r
tab <- cbind("QDA" = as.factor(predictionAnswers2),
             "nb" = as.factor(predictionAnswers),predictionAnswers==predictionAnswers2)
tab
```

```
##       QDA nb
##  [1,]   3  3 1
##  [2,]   1  1 1
##  [3,]   3  1 0
##  [4,]   1  1 1
##  [5,]   3  1 0
##  [6,]   2  2 1
##  [7,]   4  4 1
##  [8,]   2  4 0
##  [9,]   1  1 1
## [10,]   1  1 1
## [11,]   1  1 1
## [12,]   3  3 1
## [13,]   2  2 1
## [14,]   1  1 1
## [15,]   5  5 1
## [16,]   5  2 0
## [17,]   1  1 1
## [18,]   2  5 0
## [19,]   2  5 0
## [20,]   2  2 1
```