ML_Assignment_Week4

10 oktober 2017

Practical Machine Learning: Assignment Prediction

Executive summary

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). Goal of this project is to train a prediction model to predict the manner in which they did the exercise. This is the “classe” variable in the training set. The prediction model will be used to predict 20 different test cases in the test data.

The data for this project come from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Loading and Exploratory data analysis

loading

```
training <- read.csv("pml-training.csv")
testing <- read.csv("pml-testing.csv")
```

exploratory data analysis

```
dim(training)
```

```
## [1] 19622   160
```

```
# str(training)
# head(training)
# summary(training)
```

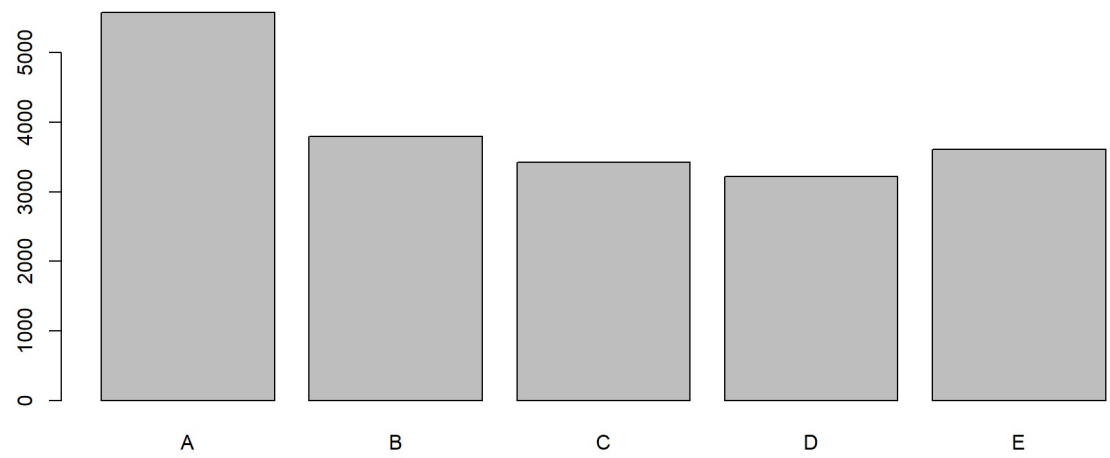
The training dataset has 160 columns (159 features) which contain a lot of NA- and empty-values. These values will be removed. Also the zero covariates and the first 5 columns, used for identification, will be removed.

```
# remove NA- and empty- values:
perc.max <- 10 # max number of rows with NA or empty-values allowed per column
row.max <- nrow(training) * perc.max/100
column.rm <- which(colSums(is.na(training) | training == "") > row.max)
training2 <- training[, -column.rm]
testing2 <- testing[, -column.rm]
# remove the zero covariates (variables which have no variability)
NZV <- nearZeroVar(training2)
training3 <- training2[ ,-NZV]
testing3 <- testing2[,-NZV]
# remove the first 5 columns with identifiers and time related features:
training4 <- training3[,-(1:5)]
testing4 <- testing3[,-(1:5)]
dim(training4)
```

```
## [1] 19622    54
```

After cleaning 54 columns are left. To visualize the dataset a barplot is created for the class variable “classe” to get a graphical representation of the class distribution. Class-A activity is the most frequently used activity.

```
plot(training4$classe)
```



Since the testing dataset provided will only be used for the quiz results generation, the training dataset will be partioned into a Training set (70% of the data) for the modeling process and a Test set (with 30% of the data) for the validations.

```
set.seed(246810)
inTrain <- createDataPartition(y=training4$classe, p = 0.7, list = FALSE)
trainSet <- training4[inTrain,]
testSet <- training4[-inTrain,]
```

Prediction model building

For this supervised multiclass classification problem two methods can be applied to train the prediction model: Random Forest and Generalized Boosted Method (GBM). These methods are commonly used for these kind of problems. They deal well with possibly correlated predictors and high dimensional datasets. Below, Random Forest is applied. Within the training partition 7-fold cross validation will be used to improve the model fit. Experimenting with the number of trees for this problem shows that the error rate does not decline a lot after 50 trees. For this reason ntree=100 will be chosen:

```
set.seed(246810)
number = 7
control.Rf <- trainControl(method="cv", number, verboseIter=FALSE)
modfit.Rf <- train(classe ~., data = trainSet, method="rf", trControl = control.R
f, ntree = 100)
modfit.Rf$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 100, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.26%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3905    1    0    0    0 0.0002560164
## B    6 2648    3    1    0 0.0037622272
## C    0    6 2390    0    0 0.0025041736
## D    0    0   11 2241    0 0.0048845471
## E    0    1    0    7 2517 0.0031683168
```

```
predRf.test <- predict(modfit.Rf, newdata=testSet)
confusionMatrix(testSet$classe, predRf.test)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    0    0    0    1
##          B    3 1136    0    0    0
##          C    0    1 1025    0    0
##          D    0    0    2  962    0
##          E    0    0    0    4 1078
##
## Overall Statistics
##
##                Accuracy : 0.9981
##                  95% CI : (0.9967, 0.9991)
##     No Information Rate : 0.2848
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9976
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9991   0.9981   0.9959   0.9991
## Specificity            0.9998   0.9994   0.9998   0.9996   0.9992
## Pos Pred Value         0.9994   0.9974   0.9990   0.9979   0.9963
## Neg Pred Value         0.9993   0.9998   0.9996   0.9992   0.9998
## Prevalence             0.2848   0.1932   0.1745   0.1641   0.1833
## Detection Rate         0.2843   0.1930   0.1742   0.1635   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9990   0.9992   0.9989   0.9977   0.9991
```

```
ose <- 1 - as.numeric(confusionMatrix(testSet$classe, predRf.test)$overall[1])
ose
```

```
## [1] 0.001869159
```

The expected out of sample error (ose) is 0.19% The accuracy is very high: 0.9981. Because of this high accuracy the Generalized Boosted Method (GBM) will not be investigated.

Applying the Random Forest model to get the quiz-answers

With the prediction model “modfit.Rf” predictions for the 20 test cases in “testing4” are made. The predictions will be used for the “Course Project Prediction Quiz Portion”.

```
predRf.testing4 <- predict(modfit.Rf, newdata = testing4)
predRf.testing4
```

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