Practical Maching Learning Coursera Assignment

Erik Hirschfeld

7 September 2019

Introduction

This document is the research for the final assignment of the Coursera course Practical Machine Learning. In this document we research the question how well people do their excercises given the information from fitness trackers.

The ultimate goal is then to predict new data given from the course website.

The document itself is created by using knitr and RStudio.

```
library(tidyverse)
#library(ggplot2)
library(data.table)
library(caret)
library(ranger)
```

Loading the Data

```
# first load the data
pth <- "D:/Online Courses/Coursera/Data Science Specialization/Practical Machine Learning/"
train <- fread(paste0(pth, "pml-training.csv"), data.table = FALSE, stringsAsFactors = TRUE)
test  <- fread(paste0(pth, "pml-testing.csv"), data.table = FALSE, stringsAsFactors = TRUE)
```

Examine the data

First look at the data and the structure of the data:

```
summary(train)
```

We can see some columns which have all the information completly missing and other columns which are completly zero and do not have any variance. We want to drop these columns.

```
drop_vars <- c("V1", "kurtosis_yaw_forearm", "skewness_yaw_forearm", "skewness_yaw_dumbbell", "kurtosis
```

Next we are checking if there are more columns with many missing and drop them.

```
missing_features <- colSums(is.na(train))
missing_features <- missing_features[missing_features > ncol(train) * 0.9]

drop_vars <- unique(c(drop_vars, names(missing_features)))

train %>%
    select(-one_of(drop_vars)) ->
    train

test %>%
    select(-one_of(drop_vars)) ->
  test
```

Modelling

Now we are starting to model the data in a first try with Random Forests via the ranger package. We start with a random forect model because normally random forest is a well performing model. The random forest method creates subsamples (e.g. 300) of the available data and columns and for each subset a decision tree is calculated. To get the final prediction the majority vote of the 300 trees are used. In case random forest will not work we good try other models like extreme gradient boosted trees.

```
train_control <- trainControl(method = "cv", number = 5)
mdl_ranger_caret <- train(classe ~ ., data = train, trControl = train_control,
                                                    method="ranger")
mdl_ranger_caret
```

```
## Random Forest
##
## 19622 samples
##    58 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 15697, 15695, 15699, 15699, 15698
## Resampling results across tuning parameters:
##
##   mtry  splitrule   Accuracy   Kappa
##    2    gini        0.9902662  0.9876852
##    2    extratrees  0.9666192  0.9577180
##   41    gini        0.9995412  0.9994197
##   41    extratrees  0.9989296  0.9986461
##   80    gini        0.9992865  0.9990975
##   80    extratrees  0.9991336  0.9989041
##
## Tuning parameter 'min.node.size' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 41, splitrule = gini
##  and min.node.size = 1.
```

We can already see that the accruacy is really good on the k-fold cross validation and we will use the model to predict our test data. The final model selected is the mtry=41 with the gini split rule. Tis model has an accruacy of 99.9% on the k-fold cross validation, so the accruacy on the test data should be over 99%.

Predicting new data

```
pred_test <- predict(mdl_ranger_caret, newdata = test)
pred_test
```

```
##  [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
```

Conclusion

After putting in the predicted values on the Coursera webpage we can see that all the predictions are correct. Given also all model statistics, especially the accruacy from the k-folf cross validation of over 99% we can conclude that we have a good model to predict the classe variable.