Course Project Prediction

Judith C. Koops

4-10-2021

Data

Loading data and packages

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
training <- read.csv("pml-training.csv", sep = ",")
```

For this assignment I have chosen random forest as the analytical tool to establish which predictors best predict which weight lifting activity (classe) was performed. I chose this model for its high accuracy performance.

The following classes are distinguished:

- Class A – exactly according to the specification
- Class B – throwing the elbows to the front
- Class C – lifting the dumbbell only halfway
- Class D – lowering the dumbbell only halfway
- Class E – throwing the hips to the front

I have deleted the variables that are directly related with the person performing the task (like user name, timestamp etc.). By deleting this information, the model is only trained on the information of the accelerometers. Some accelerometer information were captured as a string variable, I have transformed these to numeric variables.

Random forest cannot be performed on variables which have NA's. Because NA's might be informative for the model, rather than list-wise deleteting these cases, I have set them to 0.

```
training <- training %>% select(-c("X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cv
training[,1:152] <- sapply(training[,1:152],as.numeric)
training[is.na(training)] <- 0
```

Running a random forest model on the whole dataset was too computational intensive for my device. Instead, I ran the model on 983 random observations (= 5% of total training set). I use the remaining 95% of the training dataset for cross-validation and an estimate of the out of sample error of my model.

```
set.seed(10)
inTrain <- createDataPartition(y = training$classe, p = 0.05, list = FALSE)
training_model <- training[inTrain,]
training_validation <- training[-inTrain,]

dim(training_model)
```

```
## [1] 983 153
```

```
dim(training_validation)
```

```
## [1] 18639   153
```

Descriptives

```
table(training_model$classe)
```

```
##
##   A   B   C   D   E
## 279 190 172 161 181
```

Prepare model using random forest

Using the training dataset I trained model to predict the categorical variable *classe* with any of the other variables in the training dataset. I saved the model under the name *modFit1*.

```
modFit <- train(classe ~ ., data = training_model, method = "rf", prox = TRUE)
print(modFit$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)), proximity = TRUE)
##                Type of random forest: classification
```

```
##                       Number of trees: 500
## No. of variables tried at each split: 77
##
##         OOB estimate of  error rate: 8.04%
## Confusion matrix:
##     A   B   C   D   E class.error
## A 275   1   1   2   0  0.01433692
## B   9 163  14   2   2  0.14210526
## C   1   6 162   3   0  0.05813953
## D   5   1  10 143   2  0.11180124
## E   0   8   8   4 161  0.11049724
```

I examine the cross-validation by applying the model to the remaining 95% of the testing datase, which I called the training_validation dataset. My models give an out of sample error rate of 1931 / (1931 + 16708) = 10.4%

```
predicted <- predict(modFit,training_validation)
training_validation$predCorrect <- predicted == training_validation$classe

table(training_validation$predCorrect)
```

```
##
## FALSE  TRUE
##  1931 16708
```

```
table(predicted,training_validation$classe)
```

```
##
## predicted    A    B    C    D    E
##         A 5116  338    5   42    1
##         B   38 2985  216   32   98
##         C   55  176 3013  393  139
##         D   51  106   11 2555  149
##         E   41    2    5   33 3039
```

In a final step, I use the module to predict the 20 test cases avialable in the test data.

```
testing <- read.csv("pml-testing.csv", sep = ",")
testing <- testing %>% select(-c("X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvt
testing[,1:152] <- sapply(testing[,1:152],as.numeric)
testing[is.na(testing)] <- 0

pred_test <- predict(modFit,testing)
pred_test
```

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