Prediction in Weight Lifting Exercises

Anna Huynh

1/12/2021

Synopsis

This is a project towards scientific research of human activity recognition, which is focused on discriminating between different human activities (sitting/standing/walking etc.). The approach we propose for Weight Lifting Exercises for the sake of investigating how well an activity performed by the device wearer. Therefore, we might predict the manner in which they did exercise rather than only quantify how much of a particular activity they do, i.e. sports training, clinical training and so on.

The goal of our first experiment was to assess whether we could detect mistakes in weight-lifting exercises of 06 participants in the study. In particular, the algorithm we made is eventually to predict which exercise participants took throughout 17 important indicators (let's see how we figured out 17 amongst 160 features of data-set) reported by a sensor device worn by themselves.

The write-up will walk you through the following pinpoints:

- How we build the model to learn the mapping from input to output.
- How we used cross-validation to understand how well the model will perform.
- What we think the expected out of sample error is.
- Why we made the choices.

  Eventually, we use our prediction model to forecast which exercise (class) applied in 20 different test cases, where we don't actually know the outcomes. The links are enclosed.

Training Data : https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv Testing Data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

Data is collected from the study, whereas 06 participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: - 1. Exactly according to the specification (***Class A***) - 2. Throwing the elbows to the front (***Class B***) - 3. Lifting the dumbbell only halfway (***Class C***) - 4. Lowering the dumbbell only halfway (***Class D***) - 5. Throwing the hips to the front (***Class E***)

More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

1. Getting Data

```
library(readr)

train_pml <- read_csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   user_name = col_character(),
##   cvtd_timestamp = col_character(),
##   new_window = col_character(),
##   kurtosis_roll_belt = col_character(),
##   kurtosis_picth_belt = col_character(),
##   kurtosis_yaw_belt = col_character(),
##   skewness_roll_belt = col_character(),
##   skewness_roll_belt.1 = col_character(),
##   skewness_yaw_belt = col_character(),
##   max_yaw_belt = col_character(),
##   min_yaw_belt = col_character(),
##   amplitude_yaw_belt = col_character(),
##   kurtosis_picth_arm = col_character(),
##   kurtosis_yaw_arm = col_character(),
##   skewness_pitch_arm = col_character(),
##   skewness_yaw_arm = col_character(),
##   kurtosis_yaw_dumbbell = col_character(),
##   skewness_yaw_dumbbell = col_character(),
##   kurtosis_roll_forearm = col_character(),
##   kurtosis_picth_forearm = col_character()
##   # ... with 8 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
## Warning: 182 parsing failures.
##  row               col expected  actual
              file
## 2231 kurtosis_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudfront.net/predmachlear
n/pml-training.csv'
## 2231 skewness_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudfront.net/predmachlear
n/pml-training.csv'
## 2255 kurtosis_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudfront.net/predmachlear
n/pml-training.csv'
## 2255 skewness_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudfront.net/predmachlear
n/pml-training.csv'
## 2282 kurtosis_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudfront.net/predmachlear
n/pml-training.csv'
## .... ................. ........ ....... ...................................................
...................
## See problems(...) for more details.
```

```
test_pml <- read_csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_logical(),
##   X1 = col_double(),
##   user_name = col_character(),
##   raw_timestamp_part_1 = col_double(),
##   raw_timestamp_part_2 = col_double(),
##   cvtd_timestamp = col_character(),
##   new_window = col_character(),
##   num_window = col_double(),
##   roll_belt = col_double(),
##   pitch_belt = col_double(),
##   yaw_belt = col_double(),
##   total_accel_belt = col_double(),
##   gyros_belt_x = col_double(),
##   gyros_belt_y = col_double(),
##   gyros_belt_z = col_double(),
##   accel_belt_x = col_double(),
##   accel_belt_y = col_double(),
##   accel_belt_z = col_double(),
##   magnet_belt_x = col_double(),
##   magnet_belt_y = col_double(),
##   magnet_belt_z = col_double()
##   # ... with 40 more columns
## )
## See spec(...) for full column specifications.
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v dplyr   1.0.2
## v tibble  3.0.4     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v purrr   0.3.4
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'stringr' was built under R version 4.0.3
```

```
## Warning: package 'forcats' was built under R version 4.0.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.3
```

```
head(train_pml)
```

```
## # A tibble: 6 x 160
##      X1 user_name raw_timestamp_p~ raw_timestamp_p~ cvtd_timestamp new_window
##   <dbl> <chr>                <dbl>            <dbl> <chr>          <chr>
## 1     1 carlitos        1323084231           788290 05/12/2011 11~ no
## 2     2 carlitos        1323084231           808298 05/12/2011 11~ no
## 3     3 carlitos        1323084231           820366 05/12/2011 11~ no
## 4     4 carlitos        1323084232           120339 05/12/2011 11~ no
## 5     5 carlitos        1323084232           196328 05/12/2011 11~ no
## 6     6 carlitos        1323084232           304277 05/12/2011 11~ no
## # ... with 154 more variables: num_window <dbl>, roll_belt <dbl>,
## #   pitch_belt <dbl>, yaw_belt <dbl>, total_accel_belt <dbl>,
## #   kurtosis_roll_belt <chr>, kurtosis_picth_belt <chr>,
## #   kurtosis_yaw_belt <chr>, skewness_roll_belt <chr>,
## #   skewness_roll_belt.1 <chr>, skewness_yaw_belt <chr>, max_roll_belt <dbl>,
## #   max_picth_belt <dbl>, max_yaw_belt <chr>, min_roll_belt <dbl>,
## #   min_pitch_belt <dbl>, min_yaw_belt <chr>, amplitude_roll_belt <dbl>,
## #   amplitude_pitch_belt <dbl>, amplitude_yaw_belt <chr>,
## #   var_total_accel_belt <dbl>, avg_roll_belt <dbl>, stddev_roll_belt <dbl>,
## #   var_roll_belt <dbl>, avg_pitch_belt <dbl>, stddev_pitch_belt <dbl>,
## #   var_pitch_belt <dbl>, avg_yaw_belt <dbl>, stddev_yaw_belt <dbl>,
## #   var_yaw_belt <dbl>, gyros_belt_x <dbl>, gyros_belt_y <dbl>,
## #   gyros_belt_z <dbl>, accel_belt_x <dbl>, accel_belt_y <dbl>,
```

```
## #   accel_belt_z <dbl>, magnet_belt_x <dbl>, magnet_belt_y <dbl>,
## #   magnet_belt_z <dbl>, roll_arm <dbl>, pitch_arm <dbl>, yaw_arm <dbl>,
## #   total_accel_arm <dbl>, var_accel_arm <dbl>, avg_roll_arm <dbl>,
## #   stddev_roll_arm <dbl>, var_roll_arm <dbl>, avg_pitch_arm <dbl>,
## #   stddev_pitch_arm <dbl>, var_pitch_arm <dbl>, avg_yaw_arm <dbl>,
## #   stddev_yaw_arm <dbl>, var_yaw_arm <dbl>, gyros_arm_x <dbl>,
## #   gyros_arm_y <dbl>, gyros_arm_z <dbl>, accel_arm_x <dbl>, accel_arm_y <dbl>,
## #   accel_arm_z <dbl>, magnet_arm_x <dbl>, magnet_arm_y <dbl>,
## #   magnet_arm_z <dbl>, kurtosis_roll_arm <dbl>, kurtosis_picth_arm <chr>,
## #   kurtosis_yaw_arm <chr>, skewness_roll_arm <dbl>, skewness_pitch_arm <chr>,
## #   skewness_yaw_arm <chr>, max_roll_arm <dbl>, max_picth_arm <dbl>,
## #   max_yaw_arm <dbl>, min_roll_arm <dbl>, min_pitch_arm <dbl>,
## #   min_yaw_arm <dbl>, amplitude_roll_arm <dbl>, amplitude_pitch_arm <dbl>,
## #   amplitude_yaw_arm <dbl>, roll_dumbbell <dbl>, pitch_dumbbell <dbl>,
## #   yaw_dumbbell <dbl>, kurtosis_roll_dumbbell <dbl>,
## #   kurtosis_picth_dumbbell <dbl>, kurtosis_yaw_dumbbell <chr>,
## #   skewness_roll_dumbbell <dbl>, skewness_pitch_dumbbell <dbl>,
## #   skewness_yaw_dumbbell <chr>, max_roll_dumbbell <dbl>,
## #   max_picth_dumbbell <dbl>, max_yaw_dumbbell <dbl>, min_roll_dumbbell <dbl>,
## #   min_pitch_dumbbell <dbl>, min_yaw_dumbbell <dbl>,
## #   amplitude_roll_dumbbell <dbl>, amplitude_pitch_dumbbell <dbl>,
## #   amplitude_yaw_dumbbell <dbl>, total_accel_dumbbell <dbl>,
## #   var_accel_dumbbell <dbl>, avg_roll_dumbbell <dbl>,
## #   stddev_roll_dumbbell <dbl>, var_roll_dumbbell <dbl>, ...
```

```
dim(train_pml)
```

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

```
dim(test_pml)
```

```
## [1]  20 160
```

2. Exploratory Data Analysis

2.1. Missing Values

```
# Compute total missing values in each column
data.frame(colSums(is.na(train_pml)))[1:20,]
```

```
##  [1]     0     0     0     0     0     0     0     0     0     0     0 19216
## [13] 19216 19216 19216 19216 19216 19216 19216 19216
```

```
library(naniar)
```

```
## Warning: package 'naniar' was built under R version 4.0.3
```

```
# Plot missing data
train_pml %>%
  slice(1:1000) %>%
  vis_miss()
```

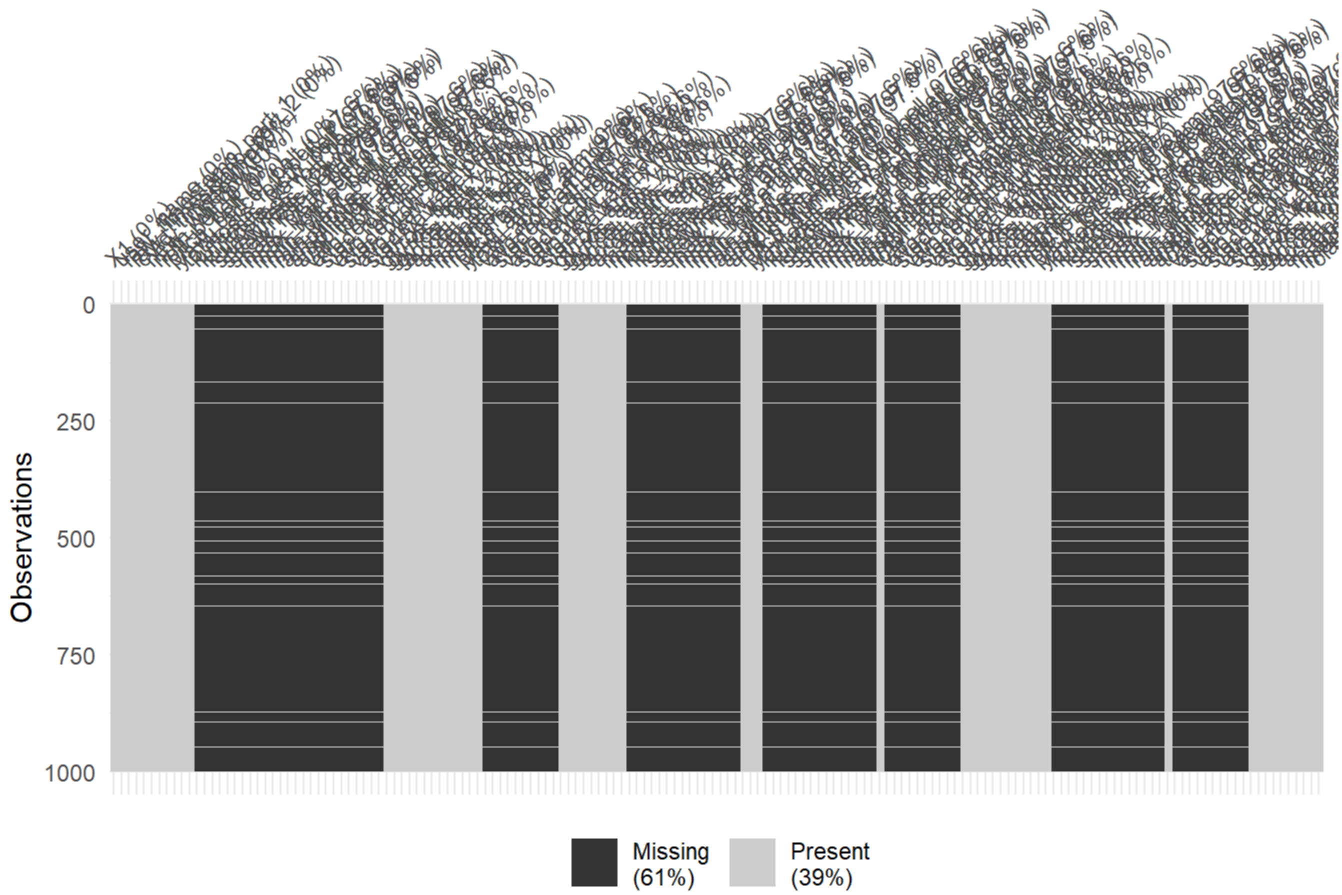


Figure 01: Plot of missing values tells us of an imbalanced data-set

2.2. How have the number of specifications (“classe”) changed per type of exercises?

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.0.3
```

```
# Need to make a new transformed data-set for this visualization

(
  classe_table <- train_pml %>%
    count(classe = factor(classe)) %>%
    mutate(pct = prop.table(n)) %>%
    arrange(-pct) %>%
    tibble()
)
```

```
## # A tibble: 5 x 3
```

```
##   classe     n   pct
##   <fct>  <int> <dbl>
## 1 A       5580 0.284
## 2 B       3797 0.194
## 3 E       3607 0.184
## 4 C       3422 0.174
## 5 D       3216 0.164
```

```
  ggplot(
  classe_table %>% filter(classe != "NA"),
  mapping = aes(
    x = reorder(classe, n),
    y = pct,
    group = 1,
    label = scales::percent(pct)
  )
) +
  theme_fivethirtyeight() +
  geom_bar(stat = "identity",
           fill = "#634832") +
  geom_text(position = position_dodge(width = 0.9),
            # move to center of bars
            hjust = -0.05,
            #Have Text just above bars
            size = 2.5) +
  labs(x = "Classes of Exercise",
       y = "Proportion of Dataset") +
  theme(axis.text.x = element_text(
    angle = 90,
    vjust = 0.5,
    hjust = 1
  )) +
  ggtitle("Classes of Exercise Listed in Weight Lifting Dataset") +
  scale_y_continuous(labels = scales::percent) +
  coord_flip()
```

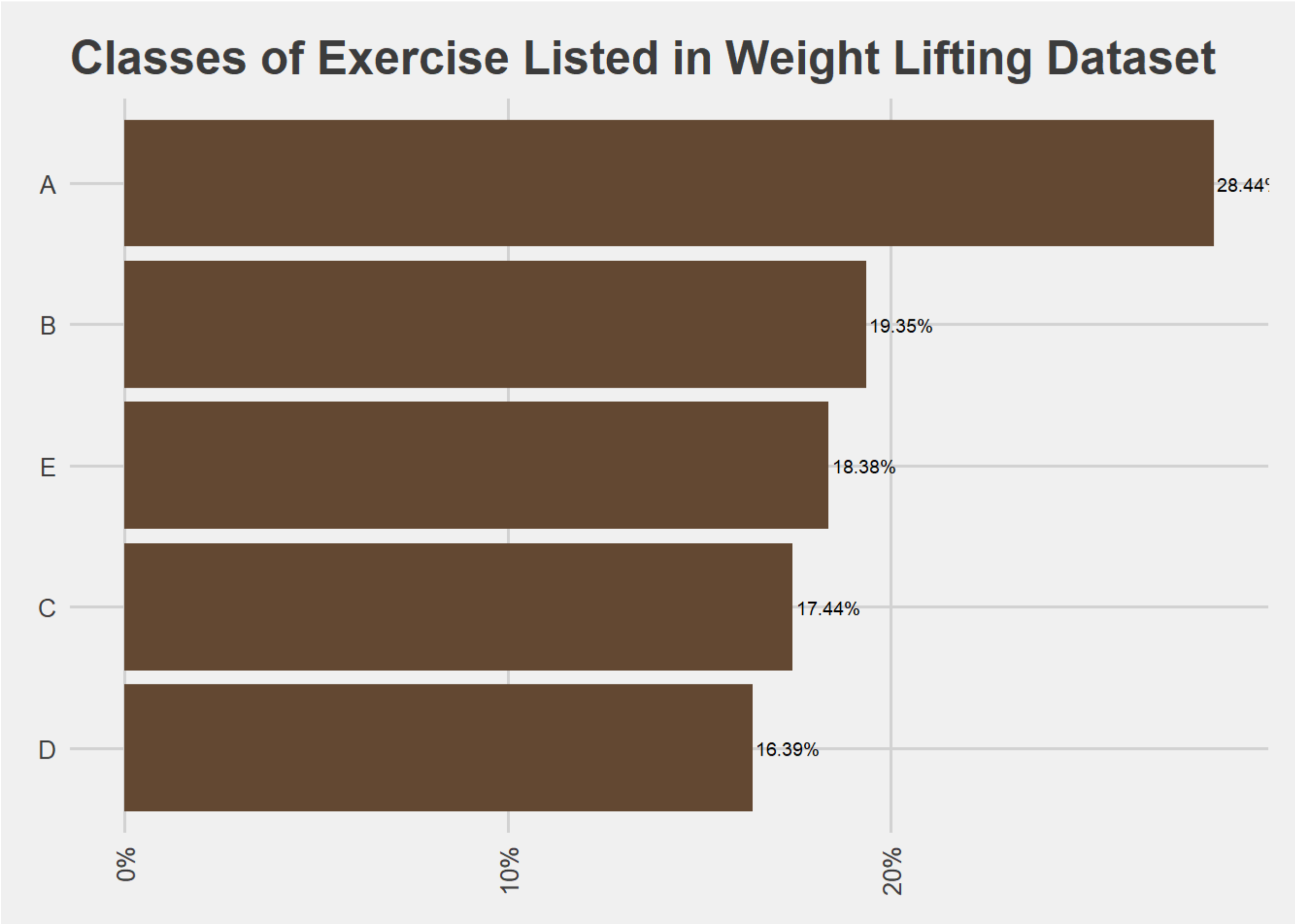


Figure 02: Class A of exercise (exactly according to the specification) dominated as compared to other classes in the data-set.

2.3. How have the number of exercises varied over days?

```
df <- train_pml %>%
  mutate(train_date = as_datetime(raw_timestamp_part_1))

df <- df %>%
  group_by(train_date) %>%
  count(classe = factor(classe)) %>%
  summarise(y = sum(n), .groups = "drop")

head(df)
```

```
## # A tibble: 6 x 2
##   train_date              y
##   <dttm>              <int>
## 1 2011-11-28 14:13:25     3
## 2 2011-11-28 14:13:26    24
## 3 2011-11-28 14:13:27    24
## 4 2011-11-28 14:13:28    24
## 5 2011-11-28 14:13:29    24
## 6 2011-11-28 14:13:30    27
```

```r
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.0.3
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
plot_ly(data = df,
        x = ~ train_date,
        y = ~ y,
        type = "scatter",
        mode = "line",
        name = "Number of Exercises") %>%
  layout(title = "Total Number of Weight Lifting Exercises per Day",
         yaxis = list(title = "Number of Exercises"),
         xaxis = list(title = "Source: Weight Lifting Dataset"))
```

Figure 03: Number of exercises varied over days

2.4. Important Features Selection

We divided into 04 groups of exercises for quantitative assessment, they are: - belt - arm - dumbbell - forearm

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# belt
```

```
train_pml_belt <- train_pml %>%
        select( classe, roll_belt, pitch_belt, yaw_belt, total_accel_belt,
               var_total_accel_belt, gyros_belt_x, gyros_belt_y, gyros_belt_z,
               accel_belt_x, accel_belt_y, accel_belt_z, magnet_belt_x,
               magnet_belt_y, magnet_belt_z ) %>%
        arrange(classe) %>%
        na.omit()


train_pml_belt$classe <- as.factor(as.character(train_pml_belt$classe))

modBelt <- randomForest(classe ~., data = train_pml_belt)
order(varImp(modBelt), decreasing=TRUE)
```

```
##  [1]  3  5  1  2 13 12 14 11  9  6 10  8  7  4
```

```
# Calculate the number of principle components needed to capture 90% of the variance
preProc_belt <- preProcess(train_pml_belt, method="pca", thresh=0.9)
preProc_belt
```

```
## Created from 406 samples and 15 variables
##
## Pre-processing:
##   - centered (14)
##   - ignored (1)
##   - principal component signal extraction (14)
##   - scaled (14)
##
## PCA needed 5 components to capture 90 percent of the variance
```

Observation: There are 09 important features extracted in the belt exercises, including: pitch_belt, total_accel_belt, classe, roll_belt, magnet_belt_x, magnet_belt_y, accel_belt_z, accel_belt_y, gyros_belt_z.

```
# arm
train_pml_arm <- train_pml %>%
        select( classe, roll_arm, pitch_arm, yaw_arm, total_accel_arm,
              gyros_arm_x, gyros_arm_y, gyros_arm_z, accel_arm_x, accel_arm_y,
              accel_arm_z, magnet_arm_x, magnet_arm_y, magnet_arm_z
              ) %>%
        arrange(classe) %>%
        na.omit()

train_pml_arm$classe <- as.factor(as.character(train_pml_arm$classe))

modArm <- randomForest(classe ~., data = train_pml_arm)
order(varImp(modArm), decreasing=TRUE)
```

```
##  [1]  1  2 12  8 10  5  3  9 11 13  6  7  4
```

```
# Calculate the number of principle components needed to capture 90% of the variance
```

```
preProc_arm <- preProcess(train_pml_arm, method="pca", thresh=0.9)
preProc_arm
```

```
## Created from 19622 samples and 14 variables
##
## Pre-processing:
##   - centered (13)
##   - ignored (1)
##   - principal component signal extraction (13)
##   - scaled (13)
##
## PCA needed 7 components to capture 90 percent of the variance
```

Observation: There are 11 important features extracted in the arm exercises, including: classe, roll_arm, magnet_arm_x, accel_arm_y, gyros_arm_z, total_accel_arm, pitch_arm, accel_arm_x, accel_arm_z, gyros_arm_x, magnet_arm_y.

```
# dumbbell
train_pml_dumbbell <- train_pml %>%
        select( classe, roll_dumbbell, pitch_dumbbell, yaw_dumbbell,
                total_accel_dumbbell, gyros_dumbbell_x, gyros_dumbbell_y,
                gyros_dumbbell_z, accel_dumbbell_x, accel_dumbbell_y,
                accel_dumbbell_z, magnet_dumbbell_x, magnet_dumbbell_y,
                magnet_dumbbell_z ) %>%
        arrange(classe) %>%
        na.omit()

train_pml_dumbbell$classe <- as.factor(as.character(train_pml_dumbbell$classe))

modDum <- randomForest(classe ~., data = train_pml_dumbbell)
order(varImp(modDum), decreasing=TRUE)
```

```
##  [1] 13 12 11 10  9  6  1  3  5  4  8  2  7
```

```
# Calculate the number of principle components needed to capture 90% of the variance
preProc_dumb <- preProcess(train_pml_dumbbell, method="pca", thresh=0.9)
preProc_dumb
```

```
## Created from 19622 samples and 14 variables
##
## Pre-processing:
##   - centered (13)
##   - ignored (1)
##   - principal component signal extraction (13)
##   - scaled (13)
##
## PCA needed 6 components to capture 90 percent of the variance
```

Observation: There are 9 important features extracted in the dumbbell, including: magnet_dumbbell_y, magnet_dumbbell_x, accel_dumbbell_z, accel_dumbbell_y, accel_dumbbell_x, gyros_dumbbell_x, classe, pitch_dumbbell, total_accel_dumbbell.

```
# forearm
train_pml_for <- train_pml %>%
        select( classe, roll_forearm, pitch_forearm, yaw_forearm,
                total_accel_forearm, gyros_forearm_x, gyros_forearm_y,
                gyros_forearm_z, accel_forearm_x, accel_forearm_y,
                accel_forearm_z, magnet_forearm_x, magnet_forearm_y,
                magnet_forearm_z ) %>%
        arrange(classe) %>%
        na.omit()

train_pml_for$classe <- as.factor(as.character(train_pml_for$classe))

modFor <- randomForest(classe ~., data = train_pml_for)
order(varImp(modFor), decreasing=TRUE)
```

```
##  [1]  2  1 13  8 10  6 11 12  3  9  7  4  5
```

```
# Calculate the number of principle components needed to capture 90% of the variance
preProc_for <- preProcess(train_pml_for, method="pca", thresh=0.9)
preProc_for
```

```
## Created from 19622 samples and 14 variables
##
## Pre-processing:
##   - centered (13)
##   - ignored (1)
##   - principal component signal extraction (13)
##   - scaled (13)
##
## PCA needed 8 components to capture 90 percent of the variance
```

Observation: There are 12 important features extracted in the forearm exercises, including: roll_forearm, classe, magnet_forearm_y, gyros_forearm_z, accel_forearm_y, magnet_forearm_x, gyros_forearm_x, pitch_forearm, accel_forearm_z, accel_forearm_x, gyros_forearm_y, yaw_forearm.

3. Build a model

3.1. Split the data and create cross-validation bootstraps.

Unify important features from 04 groups (belt, arm, dumbbell, and forearm)

```
library(caret)
set.seed(2021)

# Prepare data
sub_pml <- train_pml %>%
  select(classe, pitch_belt, total_accel_belt, roll_belt, magnet_belt_x,
         magnet_belt_y, accel_belt_z, accel_belt_y, gyros_belt_z,
         roll_arm, magnet_arm_x, accel_arm_y, gyros_arm_z, total_accel_arm,
         pitch_arm, accel_arm_x, accel_arm_z, gyros_arm_x, magnet_arm_y,
```

```
         magnet_dumbbell_y, magnet_dumbbell_x, accel_dumbbell_z, accel_dumbbell_y,
         accel_dumbbell_x, gyros_dumbbell_x, pitch_dumbbell, total_accel_dumbbell,
         roll_forearm, magnet_forearm_y, gyros_forearm_z, accel_forearm_y,
         magnet_forearm_x, gyros_forearm_x, pitch_forearm, accel_forearm_z,
         accel_forearm_x, gyros_forearm_y, yaw_forearm,
         kurtosis_roll_dumbbell ) %>%
  na.omit()


sub_pml$classe <- as.factor(as.character(sub_pml$classe))
```

```
# Reassess important features
modSub <- randomForest(classe ~., data = sub_pml)
order(varImp(modSub), decreasing=TRUE)
```

```
##  [1]  3 33 19  5 22 27 20  6  1 10  9 25 18 31 23 34 35 28 38 15 21 36 14  8 11
## [26] 24  4  2 30 17 16 13 32 26 29  7 12 37
```

```
# Calculate the number of principle components needed to capture 90% of the variance
preProc_sub <- preProcess(sub_pml, method="pca", thresh=0.9)
preProc_sub
```

```
## Created from 401 samples and 39 variables
##
## Pre-processing:
##   - centered (38)
##   - ignored (1)
##   - principal component signal extraction (38)
##   - scaled (38)
##
## PCA needed 17 components to capture 90 percent of the variance
```

We eventually picked up 17 features from ordered important variables after reassessment. They are: classe, total_accel_belt, gyros_forearm_x, magnet_arm_y, magnet_belt_x, accel_dumbbell_z, total_accel_dumbbell, magnet_dumbbell_y, magnet_belt_y, roll_arm, gyros_belt_z, gyros_dumbbell_x, gyros_arm_x, accel_forearm_y, accel_dumbbell_y, pitch_forearm, accel_forearm_z.

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3
```

```
## corrplot 0.84 loaded
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.0.3
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 4.0.3
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:plotly':
##
##     subplot
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.0.3
```

```
# Plot correlation matrix
train_pml_cor <- sub_pml %>%
  select(total_accel_belt, gyros_forearm_x, magnet_arm_y,
         magnet_belt_x, accel_dumbbell_z,
         total_accel_dumbbell, magnet_dumbbell_y, magnet_belt_y, roll_arm,
         gyros_belt_z, gyros_dumbbell_x, gyros_arm_x, accel_forearm_y,
         accel_dumbbell_y, pitch_forearm, accel_forearm_z )

pmlData <- cor(train_pml_cor)
head(round(pmlData,2))
```

```
##                        total_accel_belt gyros_forearm_x magnet_arm_y
## total_accel_belt                   1.00            0.36         0.03
## gyros_forearm_x                    0.36            1.00         0.10
```

```
## magnet_arm_y                     0.03            0.10         1.00
## magnet_belt_x                    0.29            0.50        -0.03
## accel_dumbbell_z                 0.14           -0.26        -0.13
## total_accel_dumbbell            -0.24            0.10         0.24
##                        magnet_belt_x accel_dumbbell_z total_accel_dumbbell
## total_accel_belt                0.29             0.14                -0.24
## gyros_forearm_x                 0.50            -0.26                 0.10
## magnet_arm_y                   -0.03            -0.13                 0.24
## magnet_belt_x                   1.00            -0.51                 0.33
## accel_dumbbell_z               -0.51             1.00                -0.61
## total_accel_dumbbell            0.33            -0.61                 1.00
##                        magnet_dumbbell_y magnet_belt_y roll_arm gyros_belt_z
## total_accel_belt                   -0.33         -0.27    -0.37        -0.53
## gyros_forearm_x                    -0.02         -0.07    -0.16        -0.15
## magnet_arm_y                       -0.18          0.13    -0.05        -0.13
## magnet_belt_x                       0.26         -0.05    -0.26         0.05
## accel_dumbbell_z                   -0.39         -0.41     0.36        -0.27
## total_accel_dumbbell                0.15          0.36    -0.13         0.13
##                        gyros_dumbbell_x gyros_arm_x accel_forearm_y
## total_accel_belt                  -0.04        0.10            0.05
## gyros_forearm_x                    0.01        0.01            0.33
## magnet_arm_y                       0.11       -0.02            0.05
## magnet_belt_x                     -0.07        0.11            0.33
## accel_dumbbell_z                  -0.09       -0.01           -0.14
## total_accel_dumbbell               0.21        0.00            0.18
##                        accel_dumbbell_y pitch_forearm accel_forearm_z
## total_accel_belt                  -0.31          0.14            0.11
## gyros_forearm_x                    0.15         -0.15            0.01
## magnet_arm_y                       0.07         -0.23           -0.18
## magnet_belt_x                      0.41         -0.29            0.28
## accel_dumbbell_z                  -0.67          0.25            0.26
## total_accel_dumbbell               0.77         -0.39           -0.25
```

```
cormat <- pmlData
ggcorrplot::ggcorrplot(cormat, title = "Correlation of Extracted Variables")
```

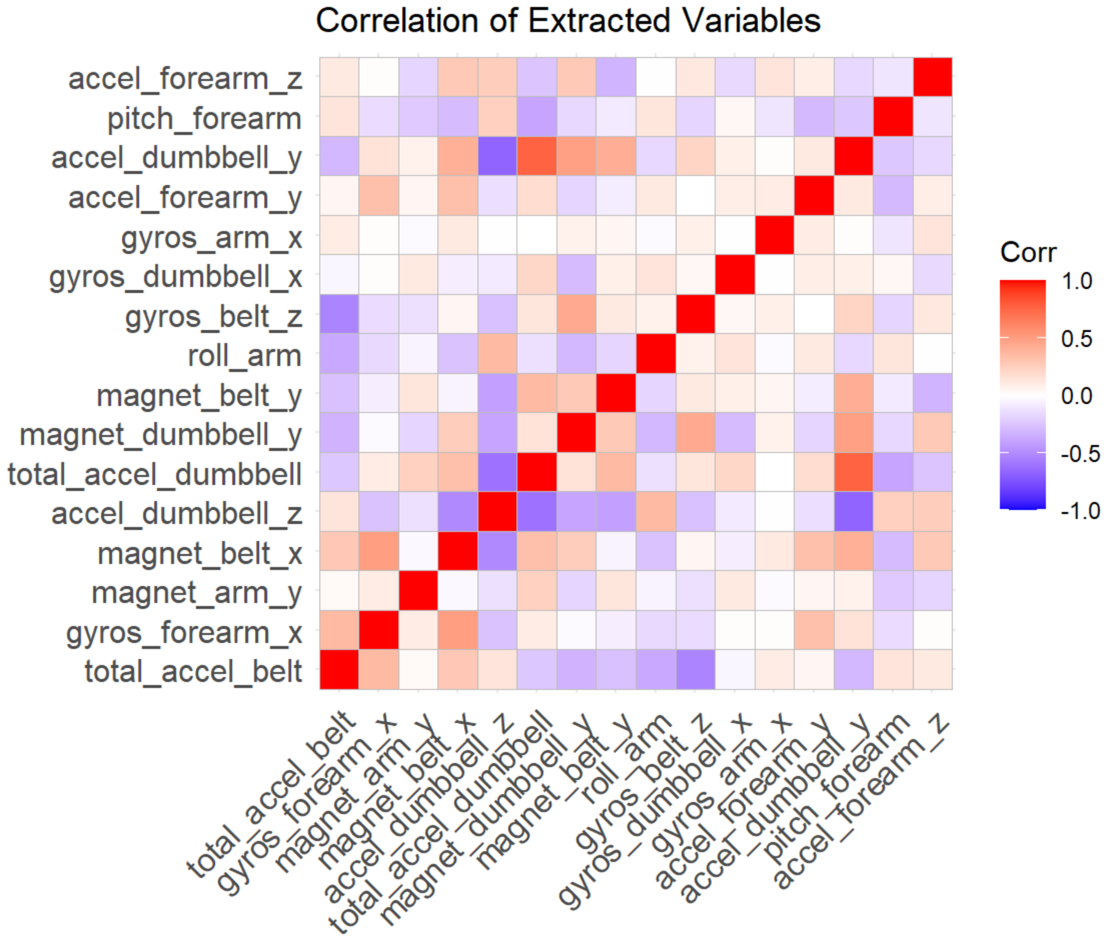
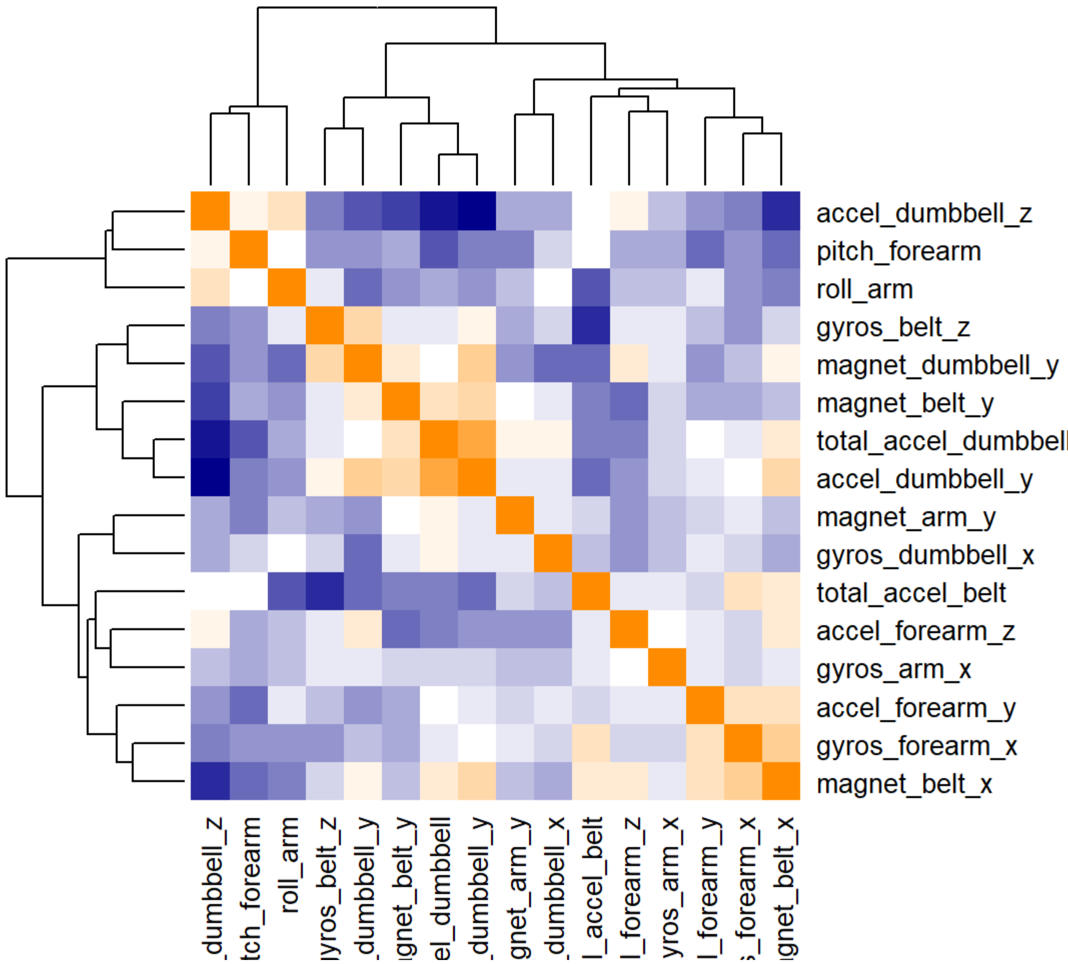


Figure 04: Correlation Matrix of important features

```
# Get some colors with Heatmap
col <- colorRampPalette(c("darkblue", "white", "darkorange"))(25)
pml_hea <- cor(train_pml_cor)
heatmap(x = pml_hea, col = col, symm = TRUE)
```



3.2. Cross-validation:

```
# Final features for training
fin_pml <- sub_pml %>%
  select(classe, total_accel_belt, gyros_forearm_x, magnet_arm_y,
         magnet_belt_x, accel_dumbbell_z,
         total_accel_dumbbell, magnet_dumbbell_y, magnet_belt_y, roll_arm,
         gyros_belt_z, gyros_dumbbell_x, gyros_arm_x, accel_forearm_y,
         accel_dumbbell_y, pitch_forearm, accel_forearm_z )
```

```
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.0.3
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --
```

```
## v broom     0.7.3      v recipes   0.1.15
## v dials     0.0.9      v rsample   0.0.8
## v infer     0.5.3      v tune      0.1.2
## v modeldata 0.1.0      v workflows 0.2.1
## v parsnip   0.1.4      v yardstick 0.0.7
```

```
## Warning: package 'broom' was built under R version 4.0.3
```

```
## Warning: package 'dials' was built under R version 4.0.3
```

```
## Warning: package 'infer' was built under R version 4.0.3
```

```
## Warning: package 'modeldata' was built under R version 4.0.3
```

```
## Warning: package 'parsnip' was built under R version 4.0.3
```

```
## Warning: package 'recipes' was built under R version 4.0.3
```

```
## Warning: package 'rsample' was built under R version 4.0.3
```

```
## Warning: package 'tune' was built under R version 4.0.3
```

```
## Warning: package 'workflows' was built under R version 4.0.3
```

```
## Warning: package 'yardstick' was built under R version 4.0.3
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x yardstick::accuracy()    masks forecast::accuracy()
## x randomForest::combine()  masks dplyr::combine()
## x scales::discard()        masks purrr::discard()
## x plotly::filter()         masks dplyr::filter(), stats::filter()
## x recipes::fixed()         masks stringr::fixed()
## x dplyr::lag()             masks stats::lag()
## x caret::lift()            masks purrr::lift()
## x randomForest::margin()   masks ggplot2::margin()
## x yardstick::precision()   masks caret::precision()
## x yardstick::recall()      masks caret::recall()
## x yardstick::sensitivity() masks caret::sensitivity()
## x yardstick::spec()        masks readr::spec()
## x yardstick::specificity() masks caret::specificity()
## x Hmisc::src()             masks dplyr::src()
## x recipes::step()          masks stats::step()
## x Hmisc::summarize()       masks dplyr::summarize()
## x parsnip::translate()     masks Hmisc::translate()
```

```
# Split the data
set.seed(2021)
pml_split <- initial_split(fin_pml, strata = classe)
pml_train <- training(pml_split) # training set
pml_test <- testing(pml_split) # validation set
```

```
# Create cross-validation bootstraps.
pml_train %>%
```

```
  count(classe)
```

```
## # A tibble: 5 x 2
##   classe     n
##   <fct>  <int>
## 1 A         81
## 2 B         60
## 3 C         53
## 4 D         50
## 5 E         59
```

```
set.seed(123)
pml_folds <- pml_train %>%
  mutate(classe = factor(classe)) %>%
  bootstraps()

pml_folds
```

```
## # Bootstrap sampling
## # A tibble: 25 x 2
##    splits            id
##    <list>            <chr>
##  1 <split [303/114]> Bootstrap01
##  2 <split [303/107]> Bootstrap02
##  3 <split [303/115]> Bootstrap03
##  4 <split [303/109]> Bootstrap04
##  5 <split [303/118]> Bootstrap05
##  6 <split [303/106]> Bootstrap06
##  7 <split [303/113]> Bootstrap07
##  8 <split [303/109]> Bootstrap08
##  9 <split [303/115]> Bootstrap09
## 10 <split [303/102]> Bootstrap10
## # ... with 15 more rows
```

Let's create a random forest model and set up a model workflow with the model and a formula pre-processor.

```
rf_spec <- rand_forest(trees = 1000) %>%
  set_mode("classification") %>%
  set_engine("ranger")

pml_wf <- workflow() %>%
  add_formula(classe ~.) %>%
  add_model(rf_spec)

pml_wf
```

```
## == Workflow ====================================================================
## Preprocessor: Formula
## Model: rand_forest()
##
```

```
## -- Preprocessor ----------------------------------------------------------------
## classe ~ .
##
## -- Model -----------------------------------------------------------------------
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   trees = 1000
##
## Computational engine: ranger
```

Let’s fit the random forest model to the bootstrap re-samples.

```
library(ranger)
```

```
## Warning: package 'ranger' was built under R version 4.0.3
```

```
##
## Attaching package: 'ranger'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
doParallel::registerDoParallel()
pml_rs <- fit_resamples(
  pml_wf,
  resamples = pml_folds,
  control = control_resamples(save_pred = TRUE)
)

pml_rs
```

```
## # Resampling results
## # Bootstrap sampling
## # A tibble: 25 x 5
##    splits           id          .metrics         .notes          .predictions
##    <list>           <chr>       <list>           <list>          <list>
##  1 <split [303/114~ Bootstrap01 <tibble [2 x 4~ <tibble [0 x ~ <tibble [114 x 9~
##  2 <split [303/107~ Bootstrap02 <tibble [2 x 4~ <tibble [0 x ~ <tibble [107 x 9~
##  3 <split [303/115~ Bootstrap03 <tibble [2 x 4~ <tibble [0 x ~ <tibble [115 x 9~
##  4 <split [303/109~ Bootstrap04 <tibble [2 x 4~ <tibble [0 x ~ <tibble [109 x 9~
##  5 <split [303/118~ Bootstrap05 <tibble [2 x 4~ <tibble [0 x ~ <tibble [118 x 9~
##  6 <split [303/106~ Bootstrap06 <tibble [2 x 4~ <tibble [0 x ~ <tibble [106 x 9~
##  7 <split [303/113~ Bootstrap07 <tibble [2 x 4~ <tibble [0 x ~ <tibble [113 x 9~
##  8 <split [303/109~ Bootstrap08 <tibble [2 x 4~ <tibble [0 x ~ <tibble [109 x 9~
##  9 <split [303/115~ Bootstrap09 <tibble [2 x 4~ <tibble [0 x ~ <tibble [115 x 9~
## 10 <split [303/102~ Bootstrap10 <tibble [2 x 4~ <tibble [0 x ~ <tibble [102 x 9~
## # ... with 15 more rows
```

3.3. Model Evaluation

```
collect_metrics(pml_rs)
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy multiclass 0.619    25 0.00926 Preprocessor1_Model1
## 2 roc_auc  hand_till  0.874    25 0.00425 Preprocessor1_Model1
```

Let's now fit to the entire training set and evaluate on the testing set.

```
pml_fit <- last_fit(pml_wf, pml_split)
collect_metrics(pml_fit)
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy multiclass     0.704 Preprocessor1_Model1
## 2 roc_auc  hand_till      0.904 Preprocessor1_Model1
```

Observation: Model's accuracy increased from 62% to 72% after fitting.

```
pml_rs %>%
  collect_predictions() %>%
  group_by(id) %>%
  ppv(classe, .pred_class)
```

```
## # A tibble: 25 x 4
##    id          .metric .estimator .estimate
##    <chr>       <chr>   <chr>          <dbl>
##  1 Bootstrap01 ppv     macro          0.593
##  2 Bootstrap02 ppv     macro          0.653
##  3 Bootstrap03 ppv     macro          0.642
##  4 Bootstrap04 ppv     macro          0.754
##  5 Bootstrap05 ppv     macro          0.625
##  6 Bootstrap06 ppv     macro          0.624
##  7 Bootstrap07 ppv     macro          0.638
##  8 Bootstrap08 ppv     macro          0.635
##  9 Bootstrap09 ppv     macro          0.608
## 10 Bootstrap10 ppv     macro          0.526
## # ... with 15 more rows
```

3.4. The expected out of sample error

```
dim(pml_train) # training test size
```

```
## [1] 303  17
```

```
collect_metrics(pml_rs)$n # number of bootstraps
```

```
## [1] 25 25
```

For training set had 303 observations, 25 boots cross validation would estimate the performance over a training size of about 290 (**the size of the expected generalization error of a training algorithm producing models out-of-samples**) which is virtually the same as the performance for training set size of 303. Thus cross-validation would not suffer from much bias. In the other words, increasing number of boots to larger values will lead to the ***bias*** in the estimate of out-of-sample (test set) accuracy ***smaller*** and the ***variance*** in the estimate of out-of-sample (test set) accuracy ***bigger***.

Next, let's compute ROC curves for each class.

```
pml_rs %>%
  collect_predictions() %>%
  group_by(id) %>%
  roc_curve(classe, .pred_A:.pred_E ) %>%
  ggplot(aes(1 - specificity, sensitivity, color = id)) +
  geom_abline(lty = 2, color = "gray80", size = 1.5) +
  geom_path(show.legend = FALSE, alpha = 0.6, size = 1.2) +
  facet_wrap(~.level, ncol = 5) +
  coord_equal()
```

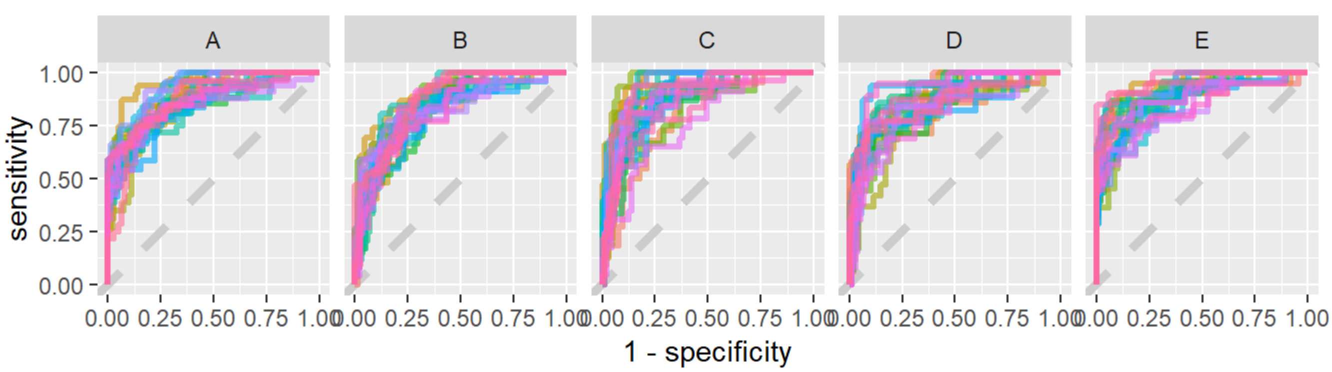


Figure 05: Plots describe ROC curve from each class of exercise

Observation: We have an ROC curve for each class and each re-sample in this plot. Notice that the points of class were easy for the model to identify.

```
pml_rs %>%
  collect_predictions() %>%
  filter(.pred_class != classe) %>%
  conf_mat(classe, .pred_class) %>%
  autoplot(type = "heatmap")
```

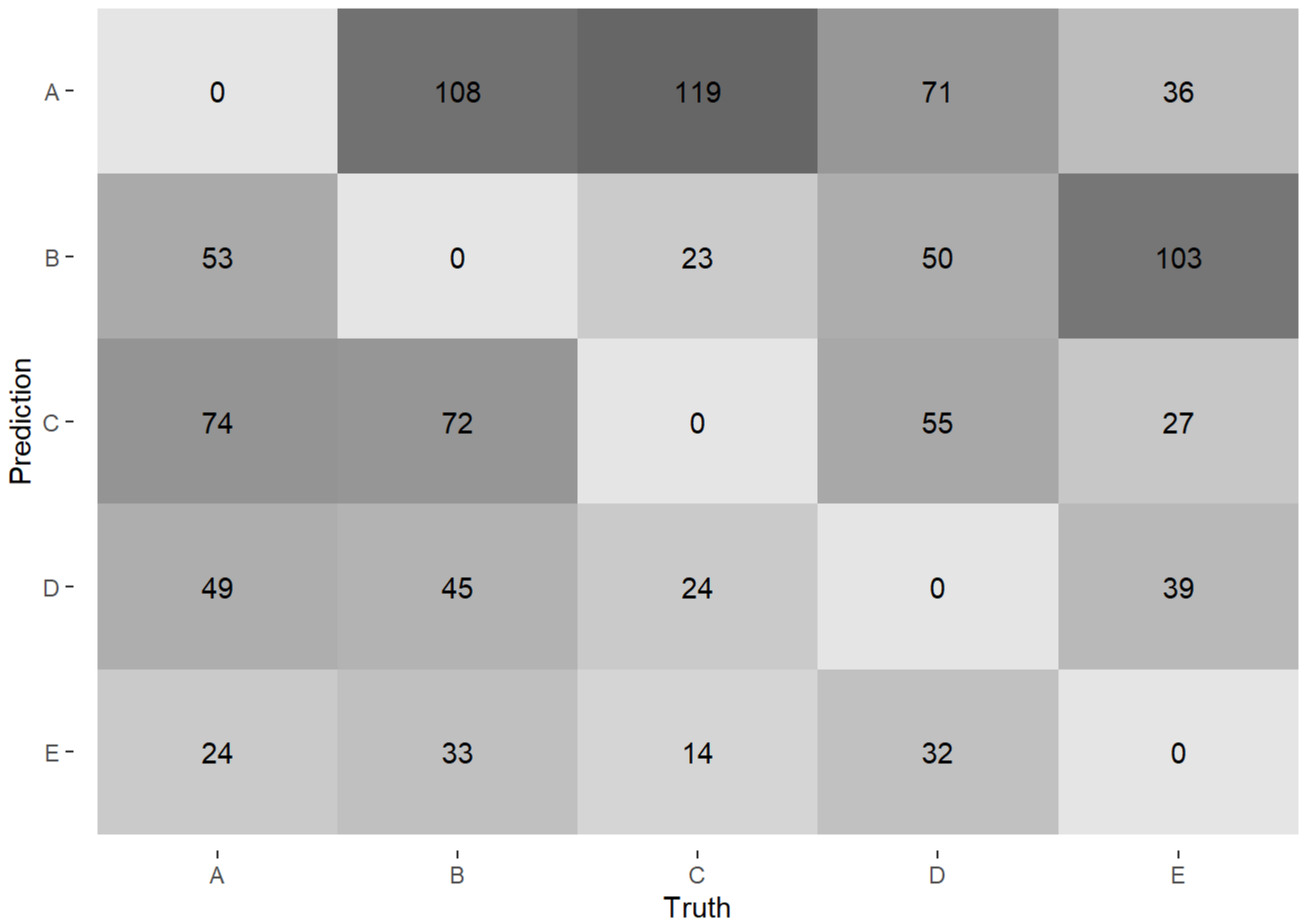


Figure 06 : Confusion Matrix of prediction and truth observations

Observation: The classes in weight lifting data-set was confused with many of the other classes, whereas class A was often confused with class C.

4. Trained model applies to validation data-set

```
# Save model
pml_wf_model <- pml_fit$.workflow[[1]]

# predict on testing set
predict(pml_wf_model, pml_test[90, ])
```

```
## # A tibble: 1 x 1
##   .pred_class
##   <fct>
## 1 C
```

5. Predict class of exercise in 20 test cases

```
predict(pml_wf_model, test_pml)
```

```
## # A tibble: 20 x 1
##    .pred_class
##    <fct>
##  1 C
##  2 B
##  3 A
##  4 A
##  5 A
##  6 C
##  7 D
##  8 B
##  9 A
## 10 A
## 11 B
## 12 C
## 13 A
## 14 A
## 15 E
## 16 B
## 17 A
## 18 B
## 19 C
## 20 B
```