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 18 important indicators (let's see how we figured out 18 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
```

```
library(naniar)
```

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

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

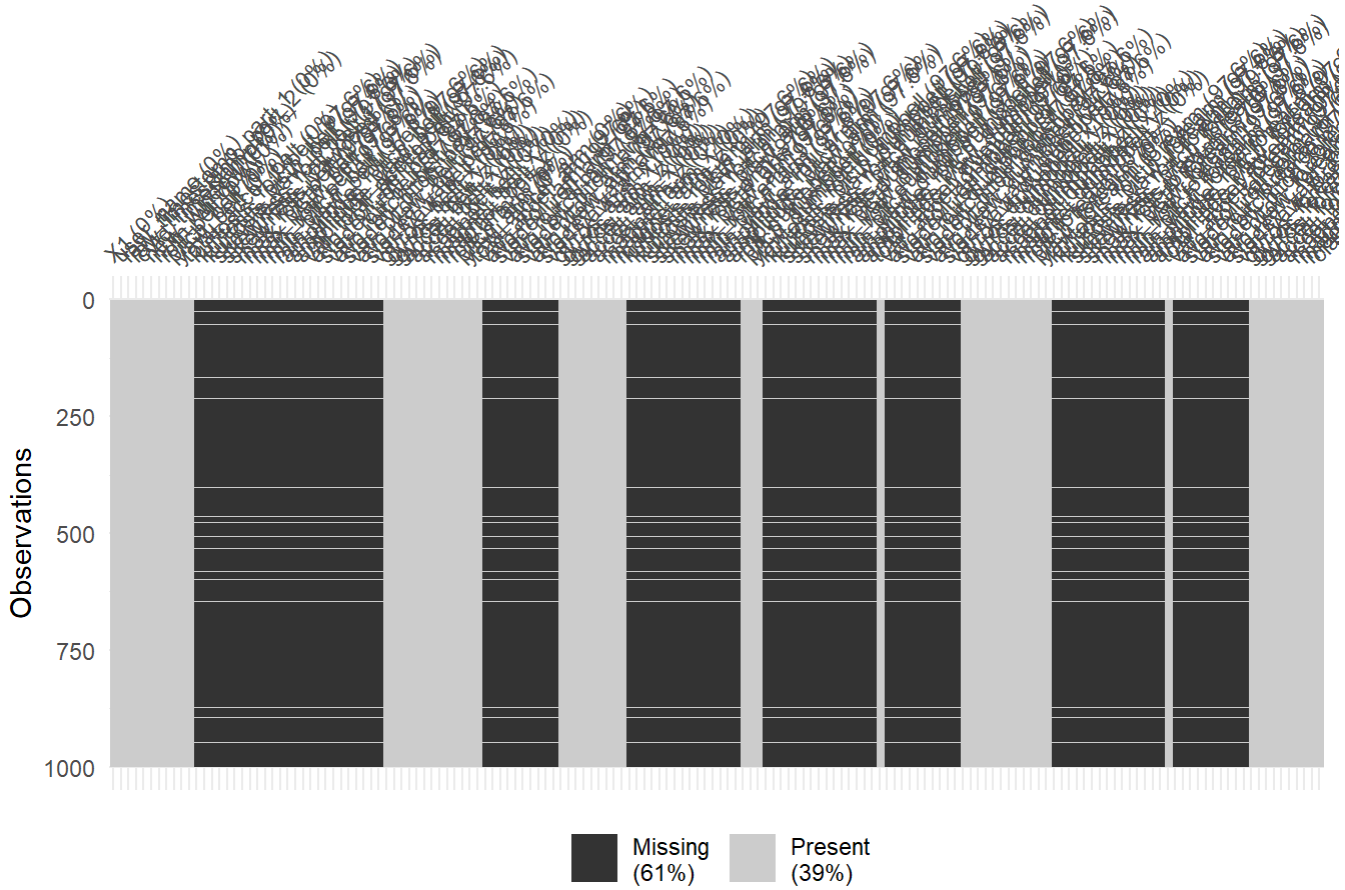

Figure 01: Plot of missing values tells us of an imbalanced data-set

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()
```

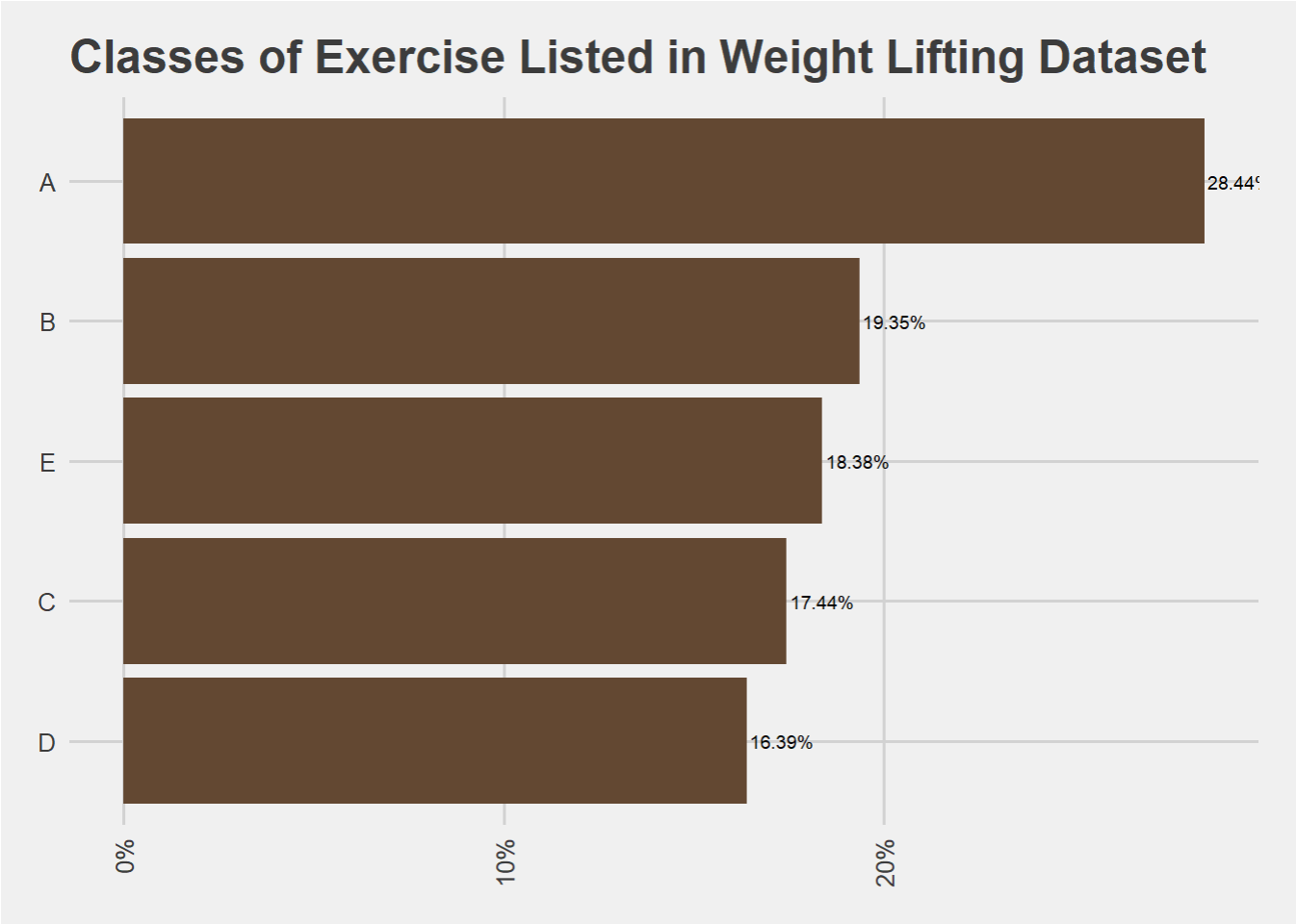

Figure 02: Class A of exercise (exactly according to the specification) dominated as compared to other classes in the data-set.

2.3. Data Transformation

```
# Drop useless features
train_pml_mod <- train_pml %>%
        select(-c(X1, user_name, raw_timestamp_part_1, raw_timestamp_part_2,
                  cvtd_timestamp, new_window, num_window) ) %>%
        arrange(classe)

# transform meaningless values
rep1 <- subset(train_pml_mod, kurtosis_picth_belt %in%
                gsub("#DIV/0!", 0, train_pml_mod$kurtosis_picth_belt) )
rep2 <- subset(rep1, kurtosis_yaw_belt %in%
                gsub("#DIV/0!", 0, train_pml_mod$kurtosis_yaw_belt) )
rep3 <- subset(rep2, skewness_roll_belt.1 %in%
                gsub("#DIV/0!", 0, train_pml_mod$skewness_roll_belt.1) )
rep4 <- subset(rep3, skewness_yaw_belt %in%
                gsub("#DIV/0!", 0, train_pml_mod$skewness_yaw_belt) )
rep5 <- subset(rep4, kurtosis_picth_arm %in%
                gsub("#DIV/0!", 0, train_pml_mod$kurtosis_picth_arm) )
rep6 <- subset(rep5, kurtosis_yaw_arm %in%
                gsub("#DIV/0!", 0, train_pml_mod$kurtosis_yaw_arm) )
rep7 <- subset(rep6, skewness_pitch_arm %in%
                gsub("#DIV/0!", 0, train_pml_mod$skewness_pitch_arm) )
```

```
rep8 <- subset(rep7, skewness_yaw_arm %in%
                 gsub("#DIV/0!", 0, train_pml_mod$skewness_yaw_arm) )
rep9 <- subset(rep8, kurtosis_yaw_dumbbell %in%
                 gsub("#DIV/0!", 0, train_pml_mod$kurtosis_yaw_dumbbell) )
rep10 <- subset(rep9, skewness_yaw_dumbbell %in%
                  gsub("#DIV/0!", 0, train_pml_mod$skewness_yaw_dumbbell) )
train_pml_cle <- subset(rep10, kurtosis_yaw_forearm %in%
             gsub("#DIV/0!", 0, train_pml_mod$kurtosis_yaw_forearm) )

train_pml_sub <- train_pml_cle %>% select(-classe)

# transform to numeric data type
train_pml_sub <- train_pml_sub[, sapply(train_pml_sub, is.numeric)]
classe <- as.factor(as.character(train_pml_cle$classe))
train_pml_com <- cbind(train_pml_sub, classe)

test_pml <- test_pml[, sapply(test_pml, is.numeric)]
```

2.5. Features Selection

After dropping features having over 90% missing values or meaningless values, our final training data owns 19,216 observations and 53 intrinsic features for modeling.

```
# Drop features having 98% missing values
sub_train <- train_pml_com %>%
  select( classe, roll_belt, yaw_belt, gyros_belt_x, gyros_belt_z, accel_belt_y,
         magnet_belt_x,
         magnet_belt_z, pitch_arm, total_accel_arm, gyros_arm_y, accel_arm_x,
         accel_arm_z, magnet_arm_y, pitch_dumbbell, gyros_dumbbell_x, gyros_dumbbell_z,
         accel_dumbbell_y, magnet_dumbbell_x,  magnet_dumbbell_z, pitch_forearm,
         total_accel_forearm, gyros_forearm_y, accel_forearm_x,  accel_forearm_z,
         magnet_forearm_y,
         pitch_belt, total_accel_belt, gyros_belt_y,  accel_belt_x,  accel_belt_z,
         magnet_belt_y, roll_arm, yaw_arm, gyros_arm_x, gyros_arm_z, accel_arm_y,
         magnet_arm_x, magnet_arm_z, roll_dumbbell, yaw_dumbbell, total_accel_dumbbell,
         gyros_dumbbell_y, accel_dumbbell_x, accel_dumbbell_z, magnet_dumbbell_y,
         roll_forearm, yaw_forearm, gyros_forearm_x, gyros_forearm_z, accel_forearm_y,
         magnet_forearm_x, magnet_forearm_z
         )

dim(sub_train)
```

```
## [1] 19216    53
```

```
which(is.na(sub_train))
```

```
## integer(0)
```

3. Build Model

3.1. Split the data into training and validation sets

```
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 scales::discard()     masks purrr::discard()
## x dplyr::filter()       masks stats::filter()
## x recipes::fixed()      masks stringr::fixed()
## x dplyr::lag()          masks stats::lag()
## x yardstick::spec()     masks readr::spec()
## x recipes::step()       masks stats::step()
```

```
# Split the data into training and validation sets
set.seed(2021)
```

```
pml_split <- initial_split(sub_train, strata = classe, prop = 3/4)
pml_train <- training(pml_split) # training set
pml_test <- testing(pml_split) # validation set
```

```
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 objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
```

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

```
# Important variables
mod_rf <- randomForest(classe ~., data = pml_train)
order(varImp(mod_rf), decreasing=TRUE)
```

```
##  [1]  1  2 20 19 26 45 46 18 17 30 39  7 31 23 44 32  4 52 37 41 43 33  6 40 24
## [26] 42 11 27 13 25 51 14 38  8 47 36 10 50 12 34 22  5 15 29 21 28  9  3 16 49
## [51] 48 35
```

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

```
## Created from 14414 samples and 53 variables
##
## Pre-processing:
##   - centered (52)
##   - ignored (1)
##   - principal component signal extraction (52)
##   - scaled (52)
##
## PCA needed 18 components to capture 90 percent of the variance
```

```
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:parsnip':
##
##     translate
```

```
## 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 of the most 18 important features
train_pml_cor <- sub_train %>%
  select(roll_belt, yaw_belt, magnet_dumbbell_x, magnet_dumbbell_z,
         magnet_forearm_y, accel_dumbbell_x, accel_dumbbell_z, magnet_dumbbell_x,
         magnet_arm_z, accel_belt_x,
         magnet_belt_z, accel_dumbbell_y, accel_belt_z, accel_forearm_z,
         accel_dumbbell_x,
         gyros_belt_z, magnet_belt_y, magnet_forearm_x)

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

```
##                   roll_belt yaw_belt magnet_dumbbell_x magnet_dumbbell_z
## roll_belt              1.00     0.82              0.31             -0.50
## yaw_belt               0.82     1.00             -0.03             -0.22
## magnet_dumbbell_x      0.31    -0.03              1.00             -0.17
## magnet_dumbbell_z     -0.50    -0.22             -0.17              1.00
## magnet_forearm_y       0.03     0.04             -0.04             -0.04
## accel_dumbbell_x       0.22     0.05              0.43              0.05
##                   magnet_forearm_y accel_dumbbell_x accel_dumbbell_z
## roll_belt                     0.03             0.22             0.10
## yaw_belt                      0.04             0.05            -0.23
## magnet_dumbbell_x            -0.04             0.43             0.53
## magnet_dumbbell_z            -0.04             0.05             0.03
## magnet_forearm_y              1.00            -0.17            -0.09
## accel_dumbbell_x             -0.17             1.00             0.68
##                   magnet_arm_z accel_belt_x magnet_belt_z accel_dumbbell_y
## roll_belt                 0.02         0.26         -0.07            -0.26
## yaw_belt                  0.02         0.71          0.09             0.06
## magnet_dumbbell_x        -0.04        -0.48         -0.29            -0.27
## magnet_dumbbell_z        -0.09         0.25         -0.31             0.23
## magnet_forearm_y          0.12         0.04          0.04             0.04
## accel_dumbbell_x         -0.09        -0.14         -0.24            -0.41
##                   accel_belt_z accel_forearm_z gyros_belt_z magnet_belt_y
```

```
## roll_belt                -0.99            0.08        -0.46         -0.21
## yaw_belt                 -0.78            0.17        -0.27         -0.06
## magnet_dumbbell_x        -0.35           -0.19        -0.50         -0.24
## magnet_dumbbell_z         0.50            0.61         0.26         -0.19
## magnet_forearm_y         -0.03            0.03         0.01          0.00
## accel_dumbbell_x         -0.27            0.27        -0.23         -0.37
##                    magnet_forearm_x
## roll_belt                     -0.19
## yaw_belt                      -0.09
## magnet_dumbbell_x             -0.10
## magnet_dumbbell_z              0.24
## magnet_forearm_y              -0.30
## accel_dumbbell_x              -0.03
```

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

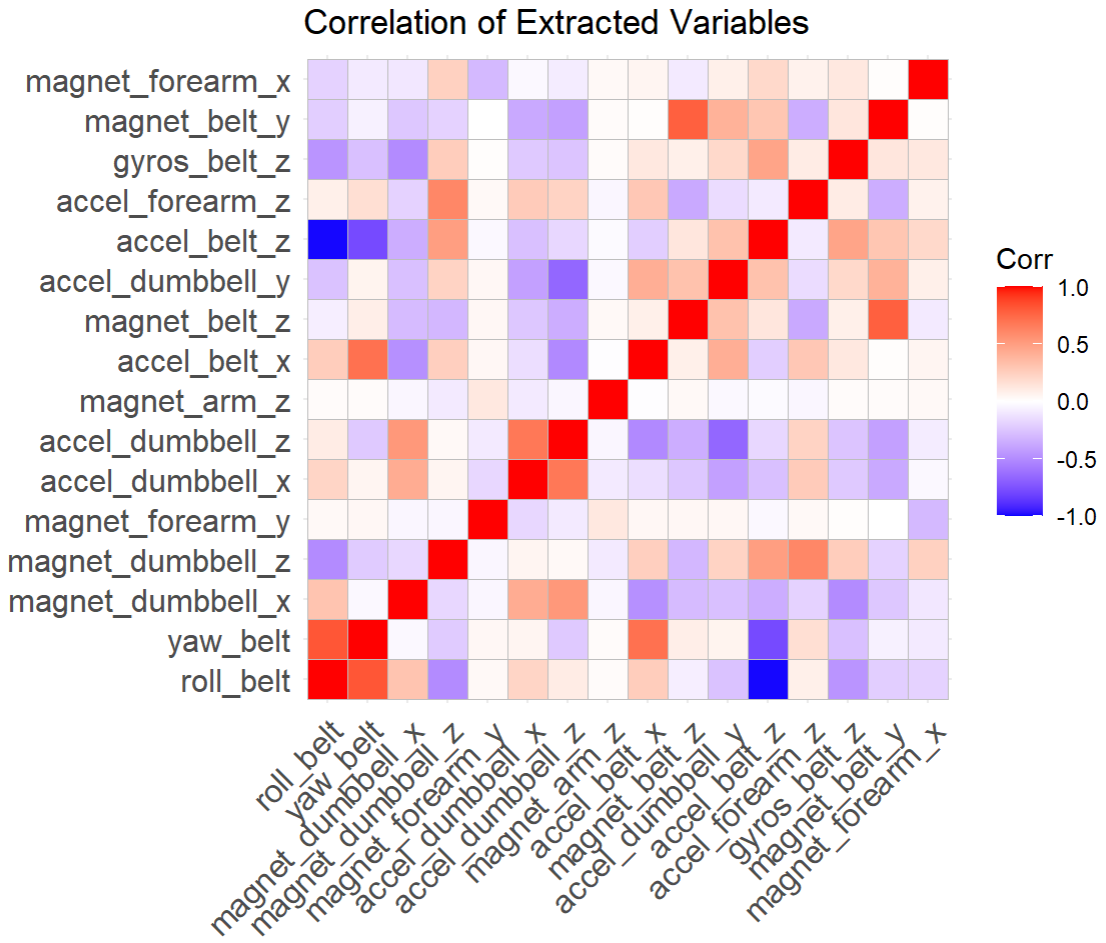

Figure 03: Correlation Matrix of the most 18 important features

3.2. Cross-validation

```
# Create cross-validation bootstraps.
pml_train %>%
  count(classe)
```

```
##   classe    n
## 1      A 4104
## 2      B 2789
## 3      C 2514
## 4      D 2361
## 5      E 2646
```

```
set.seed(123)
pml_folds <- pml_train %>%
  mutate(classe = factor(classe)) %>%
  bootstraps(5)

pml_folds
```

```
## # Bootstrap sampling
## # A tibble: 5 x 2
##   splits               id
##   <list>               <chr>
## 1 <split [14.4K/5.3K]> Bootstrap1
## 2 <split [14.4K/5.3K]> Bootstrap2
## 3 <split [14.4K/5.3K]> Bootstrap3
## 4 <split [14.4K/5.3K]> Bootstrap4
## 5 <split [14.4K/5.3K]> Bootstrap5
```

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 = 250) %>%
  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 = 250
##
## 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: 5 x 5
##   splits             id         .metrics       .notes          .predictions
##   <list>             <chr>      <list>         <list>          <list>
## 1 <split [14.4K/5.3~ Bootstrap1 <tibble [2 x ~ <tibble [0 x ~ <tibble [5,317 x ~
## 2 <split [14.4K/5.3~ Bootstrap2 <tibble [2 x ~ <tibble [0 x ~ <tibble [5,322 x ~
## 3 <split [14.4K/5.3~ Bootstrap3 <tibble [2 x ~ <tibble [0 x ~ <tibble [5,255 x ~
## 4 <split [14.4K/5.3~ Bootstrap4 <tibble [2 x ~ <tibble [0 x ~ <tibble [5,339 x ~
## 5 <split [14.4K/5.3~ Bootstrap5 <tibble [2 x ~ <tibble [0 x ~ <tibble [5,309 x ~
```

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 0.989     5 0.000818  Preprocessor1_Model1
## 2 roc_auc  hand_till  1.00      5 0.0000479 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     0.996 Preprocessor1_Model1
## 2 roc_auc  hand_till      1.00  Preprocessor1_Model1
```

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

```
## # A tibble: 5 x 4
##   id         .metric .estimator .estimate
##   <chr>      <chr>   <chr>          <dbl>
## 1 Bootstrap1 ppv     macro          0.986
## 2 Bootstrap2 ppv     macro          0.988
## 3 Bootstrap3 ppv     macro          0.989
## 4 Bootstrap4 ppv     macro          0.990
## 5 Bootstrap5 ppv     macro          0.987
```

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()
```

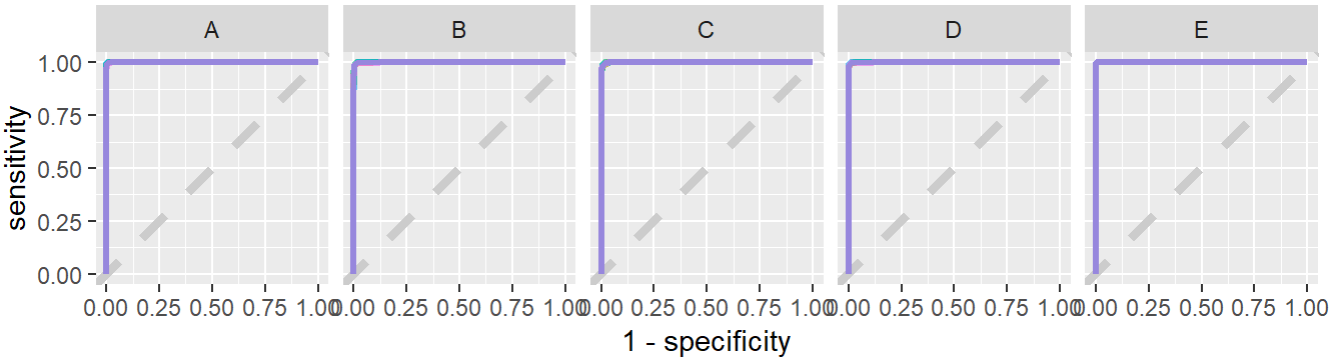

Figure 04: Plots describe ROC curve from each class of exercise

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")
```

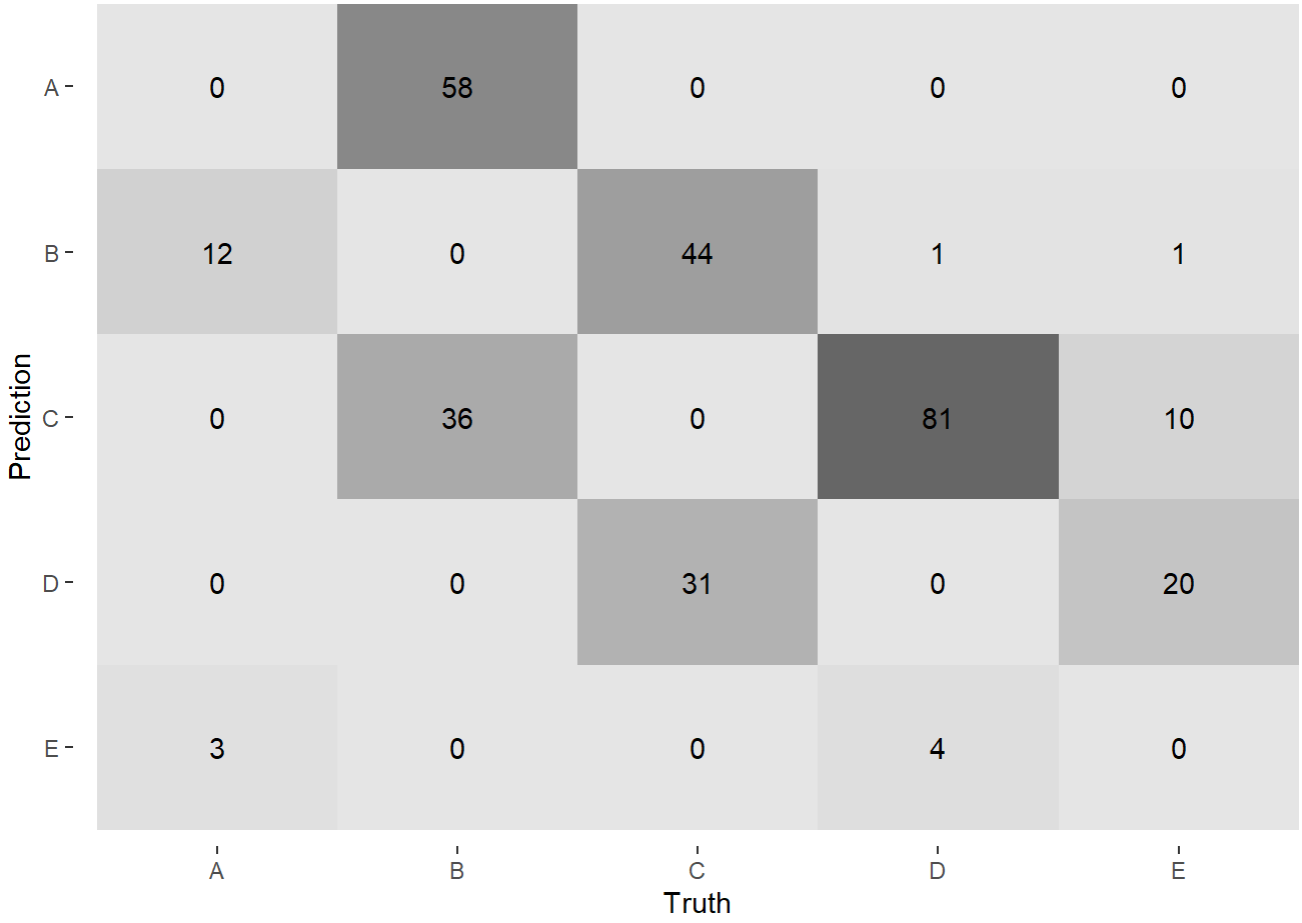

Figure 05: Confusion Matrix of prediction and truth observations

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

4. Trained model applied to validation data-set & expected out-of-sample error

4.1. Cross-validation on validation dataset

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

# predict on testing set
predict(pml_wf_model, pml_test[70, ])
```

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

4.2. Out-of-sample-error

```
control_rf <- trainControl(method = "cv", 5)
model_rf <- train(classe ~ ., data = pml_train, method="rf",
```

```
                  trControl=control_rf, ntree=250)
model_rf
```

```
## Random Forest
##
## 14414 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11530, 11531, 11531, 11531, 11533
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9911197  0.9887646
##   27    0.9907728  0.9883267
##   52    0.9856388  0.9818320
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
predict_rf <- predict(model_rf, pml_test )

confusionMatrix(pml_test$classe, predict_rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1366    1    0    0    0
##          B    2  927    0    0    0
##          C    0    5  833    0    0
##          D    0    0    6  779    1
##          E    0    0    0    2  880
##
## Overall Statistics
##
##                Accuracy : 0.9965
##                  95% CI : (0.9943, 0.9979)
##     No Information Rate : 0.2849
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9955
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9985   0.9936   0.9928   0.9974   0.9989
## Specificity            0.9997   0.9995   0.9987   0.9983   0.9995
```

```
## Pos Pred Value         0.9993   0.9978   0.9940   0.9911   0.9977
## Neg Pred Value         0.9994   0.9985   0.9985   0.9995   0.9997
## Prevalence             0.2849   0.1943   0.1747   0.1626   0.1835
## Detection Rate         0.2845   0.1930   0.1735   0.1622   0.1833
## Detection Prevalence   0.2847   0.1935   0.1745   0.1637   0.1837
## Balanced Accuracy      0.9991   0.9965   0.9958   0.9978   0.9992
```

```
# Out-of-sample-error in validation set
OOSE <- 1 - as.numeric(confusionMatrix(pml_test$classe, predict_rf)$overall[1])
OOSE
```

```
## [1] 0.003540192
```

Observation: Expected out-of-sample-error is 0.3% when model demonstrated 99.71% in accuracy.

5. Predict class of exercise in 20 test cases

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

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