Practical Machine Learning Final Project

Executive Summary

In this project, we want to predict which type (A~E) of the weight-lifting did the participant perform using all of the variables in the data set. First, I clean the data by excluding the sequence-variable ("X") and timestamp-related variables, which are not useful in our predictions, and then I look for any near-zero-variance variables and eliminate them also. Models I tried include Classification Tree Model, Gradient Boost Model (gbm), Random Forest (rf), Latent Dirichlet Allocation (lda), Support vector machine (svm), and a model combining with RF, lda and gbm. After analyzing, I found that the combining model reaches the highest accuracy, with the 5- fold cross validation.

Background: (Copy from Coursera)

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Objective:

Predict the manner in which they did the exercise. This is the "classe" variable in the training set. Create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Loading libraries and data sets

```
library(caret)
```

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

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

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

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

```
library(ISLR)
library(ggplot2)
```

```
library(gridExtra)
library(Hmisc)
```

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

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

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

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

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

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

```
library(elasticnet)
```

```
## Loading required package: lars
```

```
## Loaded lars 1.2
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:Hmisc':
##
##     impute
```

```
library(randomForest)
```

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

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

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

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

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

```
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

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

```
pml_train <- read.csv("/Users/andrewhu/Documents/GitHub/Coursera_DataScience_JHU/Practical Machine Learn
pml_test <- read.csv("/Users/andrewhu/Documents/GitHub/Coursera_DataScience_JHU/Practical Machine Learni
```

Cleaning data

First of all, we can have a quick preview of this data set:

```
head(pml_train)
```

```
##   X user_name raw_timestamp_part_1 raw_timestamp_part_2   cvtd_timestamp
## 1 1  carlitos           1323084231               788290 05/12/2011 11:23
## 2 2  carlitos           1323084231               808298 05/12/2011 11:23
## 3 3  carlitos           1323084231               820366 05/12/2011 11:23
## 4 4  carlitos           1323084232               120339 05/12/2011 11:23
## 5 5  carlitos           1323084232               196328 05/12/2011 11:23
## 6 6  carlitos           1323084232               304277 05/12/2011 11:23
##   new_window num_window roll_belt pitch_belt yaw_belt total_accel_belt
## 1         no         11      1.41       8.07    -94.4                3
## 2         no         11      1.41       8.07    -94.4                3
## 3         no         11      1.42       8.07    -94.4                3
## 4         no         12      1.48       8.05    -94.4                3
## 5         no         12      1.48       8.07    -94.4                3
## 6         no         12      1.45       8.06    -94.4                3
##   kurtosis_roll_belt kurtosis_picth_belt kurtosis_yaw_belt
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_roll_belt skewness_roll_belt.1 skewness_yaw_belt max_roll_belt
## 1                                                                      NA
## 2                                                                      NA
## 3                                                                      NA
## 4                                                                      NA
## 5                                                                      NA
## 6                                                                      NA
##   max_picth_belt max_yaw_belt min_roll_belt min_pitch_belt min_yaw_belt
## 1             NA                         NA             NA
## 2             NA                         NA             NA
## 3             NA                         NA             NA
## 4             NA                         NA             NA
## 5             NA                         NA             NA
## 6             NA                         NA             NA
##   amplitude_roll_belt amplitude_pitch_belt amplitude_yaw_belt
## 1                  NA                   NA
## 2                  NA                   NA
## 3                  NA                   NA
## 4                  NA                   NA
## 5                  NA                   NA
```

```
## 6                  NA                   NA
##   var_total_accel_belt avg_roll_belt stddev_roll_belt var_roll_belt
## 1                   NA            NA               NA            NA
## 2                   NA            NA               NA            NA
## 3                   NA            NA               NA            NA
## 4                   NA            NA               NA            NA
## 5                   NA            NA               NA            NA
## 6                   NA            NA               NA            NA
##   avg_pitch_belt stddev_pitch_belt var_pitch_belt avg_yaw_belt
## 1             NA                NA             NA           NA
## 2             NA                NA             NA           NA
## 3             NA                NA             NA           NA
## 4             NA                NA             NA           NA
## 5             NA                NA             NA           NA
## 6             NA                NA             NA           NA
##   stddev_yaw_belt var_yaw_belt gyros_belt_x gyros_belt_y gyros_belt_z
## 1              NA           NA         0.00         0.00        -0.02
## 2              NA           NA         0.02         0.00        -0.02
## 3              NA           NA         0.00         0.00        -0.02
## 4              NA           NA         0.02         0.00        -0.03
## 5              NA           NA         0.02         0.02        -0.02
## 6              NA           NA         0.02         0.00        -0.02
##   accel_belt_x accel_belt_y accel_belt_z magnet_belt_x magnet_belt_y
## 1          -21            4           22            -3           599
## 2          -22            4           22            -7           608
## 3          -20            5           23            -2           600
## 4          -22            3           21            -6           604
## 5          -21            2           24            -6           600
## 6          -21            4           21             0           603
##   magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm var_accel_arm
## 1          -313     -128      22.5    -161              34            NA
## 2          -311     -128      22.5    -161              34            NA
## 3          -305     -128      22.5    -161              34            NA
## 4          -310     -128      22.1    -161              34            NA
## 5          -302     -128      22.1    -161              34            NA
## 6          -312     -128      22.0    -161              34            NA
##   avg_roll_arm stddev_roll_arm var_roll_arm avg_pitch_arm stddev_pitch_arm
## 1           NA              NA           NA            NA               NA
## 2           NA              NA           NA            NA               NA
## 3           NA              NA           NA            NA               NA
## 4           NA              NA           NA            NA               NA
## 5           NA              NA           NA            NA               NA
## 6           NA              NA           NA            NA               NA
##   var_pitch_arm avg_yaw_arm stddev_yaw_arm var_yaw_arm gyros_arm_x
## 1            NA          NA             NA          NA        0.00
## 2            NA          NA             NA          NA        0.02
## 3            NA          NA             NA          NA        0.02
## 4            NA          NA             NA          NA        0.02
## 5            NA          NA             NA          NA        0.00
## 6            NA          NA             NA          NA        0.02
##   gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z magnet_arm_x
## 1        0.00       -0.02        -288         109        -123         -368
## 2       -0.02       -0.02        -290         110        -125         -369
## 3       -0.02       -0.02        -289         110        -126         -368
```

```
## 4       -0.03        0.02        -289         111        -123         -372
## 5       -0.03        0.00        -289         111        -123         -374
## 6       -0.03        0.00        -289         111        -122         -369
##   magnet_arm_y magnet_arm_z kurtosis_roll_arm kurtosis_picth_arm
## 1          337          516
## 2          337          513
## 3          344          513
## 4          344          512
## 5          337          506
## 6          342          513
##   kurtosis_yaw_arm skewness_roll_arm skewness_pitch_arm skewness_yaw_arm
## 1
## 2
## 3
## 4
## 5
## 6
##   max_roll_arm max_picth_arm max_yaw_arm min_roll_arm min_pitch_arm
## 1           NA            NA          NA           NA            NA
## 2           NA            NA          NA           NA            NA
## 3           NA            NA          NA           NA            NA
## 4           NA            NA          NA           NA            NA
## 5           NA            NA          NA           NA            NA
## 6           NA            NA          NA           NA            NA
##   min_yaw_arm amplitude_roll_arm amplitude_pitch_arm amplitude_yaw_arm
## 1          NA                NA                 NA               NA
## 2          NA                NA                 NA               NA
## 3          NA                NA                 NA               NA
## 4          NA                NA                 NA               NA
## 5          NA                NA                 NA               NA
## 6          NA                NA                 NA               NA
##   roll_dumbbell pitch_dumbbell yaw_dumbbell kurtosis_roll_dumbbell
## 1      13.05217      -70.49400    -84.87394
## 2      13.13074      -70.63751    -84.71065
## 3      12.85075      -70.27812    -85.14078
## 4      13.43120      -70.39379    -84.87363
## 5      13.37872      -70.42856    -84.85306
## 6      13.38246      -70.81759    -84.46500
##   kurtosis_picth_dumbbell kurtosis_yaw_dumbbell skewness_roll_dumbbell
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_dumbbell skewness_yaw_dumbbell max_roll_dumbbell
## 1                                                              NA
## 2                                                              NA
## 3                                                              NA
## 4                                                              NA
## 5                                                              NA
## 6                                                              NA
##   max_picth_dumbbell max_yaw_dumbbell min_roll_dumbbell min_pitch_dumbbell
## 1                 NA                                 NA                 NA
```

```
## 2                 NA                                   NA                 NA
## 3                 NA                                   NA                 NA
## 4                 NA                                   NA                 NA
## 5                 NA                                   NA                 NA
## 6                 NA                                   NA                 NA
##   min_yaw_dumbbell amplitude_roll_dumbbell amplitude_pitch_dumbbell
## 1                                       NA                       NA
## 2                                       NA                       NA
## 3                                       NA                       NA
## 4                                       NA                       NA
## 5                                       NA                       NA
## 6                                       NA                       NA
##   amplitude_yaw_dumbbell total_accel_dumbbell var_accel_dumbbell
## 1                                         37                NA
## 2                                         37                NA
## 3                                         37                NA
## 4                                         37                NA
## 5                                         37                NA
## 6                                         37                NA
##   avg_roll_dumbbell stddev_roll_dumbbell var_roll_dumbbell
## 1                NA                   NA                NA
## 2                NA                   NA                NA
## 3                NA                   NA                NA
## 4                NA                   NA                NA
## 5                NA                   NA                NA
## 6                NA                   NA                NA
##   avg_pitch_dumbbell stddev_pitch_dumbbell var_pitch_dumbbell
## 1                 NA                    NA                 NA
## 2                 NA                    NA                 NA
## 3                 NA                    NA                 NA
## 4                 NA                    NA                 NA
## 5                 NA                    NA                 NA
## 6                 NA                    NA                 NA
##   avg_yaw_dumbbell stddev_yaw_dumbbell var_yaw_dumbbell gyros_dumbbell_x
## 1               NA                  NA               NA                0
## 2               NA                  NA               NA                0
## 3               NA                  NA               NA                0
## 4               NA                  NA               NA                0
## 5               NA                  NA               NA                0
## 6               NA                  NA               NA                0
##   gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y
## 1            -0.02             0.00             -234               47
## 2            -0.02             0.00             -233               47
## 3            -0.02             0.00             -232               46
## 4            -0.02            -0.02             -232               48
## 5            -0.02             0.00             -233               48
## 6            -0.02             0.00             -234               48
##   accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z
## 1             -271              -559               293               -65
## 2             -269              -555               296               -64
## 3             -270              -561               298               -63
## 4             -269              -552               303               -60
## 5             -270              -554               292               -68
## 6             -269              -558               294               -66
```

```
##   roll_forearm pitch_forearm yaw_forearm kurtosis_roll_forearm
## 1         28.4         -63.9       -153
## 2         28.3         -63.9       -153
## 3         28.3         -63.9       -152
## 4         28.1         -63.9       -152
## 5         28.0         -63.9       -152
## 6         27.9         -63.9       -152
##   kurtosis_picth_forearm kurtosis_yaw_forearm skewness_roll_forearm
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_forearm skewness_yaw_forearm max_roll_forearm
## 1                                                           NA
## 2                                                           NA
## 3                                                           NA
## 4                                                           NA
## 5                                                           NA
## 6                                                           NA
##   max_picth_forearm max_yaw_forearm min_roll_forearm min_pitch_forearm
## 1                NA                                NA                NA
## 2                NA                                NA                NA
## 3                NA                                NA                NA
## 4                NA                                NA                NA
## 5                NA                                NA                NA
## 6                NA                                NA                NA
##   min_yaw_forearm amplitude_roll_forearm amplitude_pitch_forearm
## 1                                      NA                     NA
## 2                                      NA                     NA
## 3                                      NA                     NA
## 4                                      NA                     NA
## 5                                      NA                     NA
## 6                                      NA                     NA
##   amplitude_yaw_forearm total_accel_forearm var_accel_forearm
## 1                                        36               NA
## 2                                        36               NA
## 3                                        36               NA
## 4                                        36               NA
## 5                                        36               NA
## 6                                        36               NA
##   avg_roll_forearm stddev_roll_forearm var_roll_forearm avg_pitch_forearm
## 1               NA                  NA               NA                NA
## 2               NA                  NA               NA                NA
## 3               NA                  NA               NA                NA
## 4               NA                  NA               NA                NA
## 5               NA                  NA               NA                NA
## 6               NA                  NA               NA                NA
##   stddev_pitch_forearm var_pitch_forearm avg_yaw_forearm
## 1                   NA                NA              NA
## 2                   NA                NA              NA
## 3                   NA                NA              NA
## 4                   NA                NA              NA
```

```
## 5                   NA                NA              NA
## 6                   NA                NA              NA
##   stddev_yaw_forearm var_yaw_forearm gyros_forearm_x gyros_forearm_y
## 1                 NA              NA            0.03            0.00
## 2                 NA              NA            0.02            0.00
## 3                 NA              NA            0.03           -0.02
## 4                 NA              NA            0.02           -0.02
## 5                 NA              NA            0.02            0.00
## 6                 NA              NA            0.02           -0.02
##   gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## 1           -0.02             192             203            -215
## 2           -0.02             192             203            -216
## 3            0.00             196             204            -213
## 4            0.00             189             206            -214
## 5           -0.02             189             206            -214
## 6           -0.03             193             203            -215
##   magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## 1              -17              654              476      A
## 2              -18              661              473      A
## 3              -18              658              469      A
## 4              -16              658              469      A
## 5              -17              655              473      A
## 6               -9              660              478      A
```

Then we can realize that the "X", "user_name", and "timestamp-related" variables are not useful predictors in our study. Hence, we need to remove them:

```
#Remove user-name
pml_train <- subset(pml_train, select=-c(X,cvtd_timestamp,user_name,raw_timestamp_part_1,raw_timestamp_
```

And then, there are a lot of variables having **near zero variance**. We also need to exclude them.

```
#remove near-zero var
nzv<- nearZeroVar(pml_train,saveMetrics=T)
pml_train<- pml_train[,nzv$nzv==FALSE]
```

Finally, there are still a lot of variables that contain many missing values. If a variable has more than 85% of the missing value, I decide to "kick them out".

```
#Remove NA more than 85%
pml_train <-pml_train[, colMeans(is.na(pml_train)) <=.15]
```

Now we have prepared our data. Let's devide the data set into training set and testing set: (75% to training set)

```
#Data Splitting:
set.seed(777)
inTrain <-createDataPartition(y=pml_train$classe,p=0.75, list=FALSE)

training <- pml_train[inTrain,]
testing <- pml_train[-inTrain,]

#Check the dim for traning and testing
dim(training)
```

```
## [1] 14718    54
```

```
dim(testing)
```

```
## [1] 4904   54
```

Fit Models:

Now we can try to train our model with the training data set.

First, we set a standard of 5 folds for our cross-validation:

```
#Fit-Control parameters
set.seed(777)
fitControl <- trainControl(method = "repeatedcv",
                           number = 5,
                           repeats = 1)
```

** Classification Tree **:

```
#classification tree
set.seed(777)
fit_rpart <- train(classe ~ ., preProcess= c("center","scale"), method="rpart",data=training) #Note:To


#Print our model
print(fit_rpart)
```

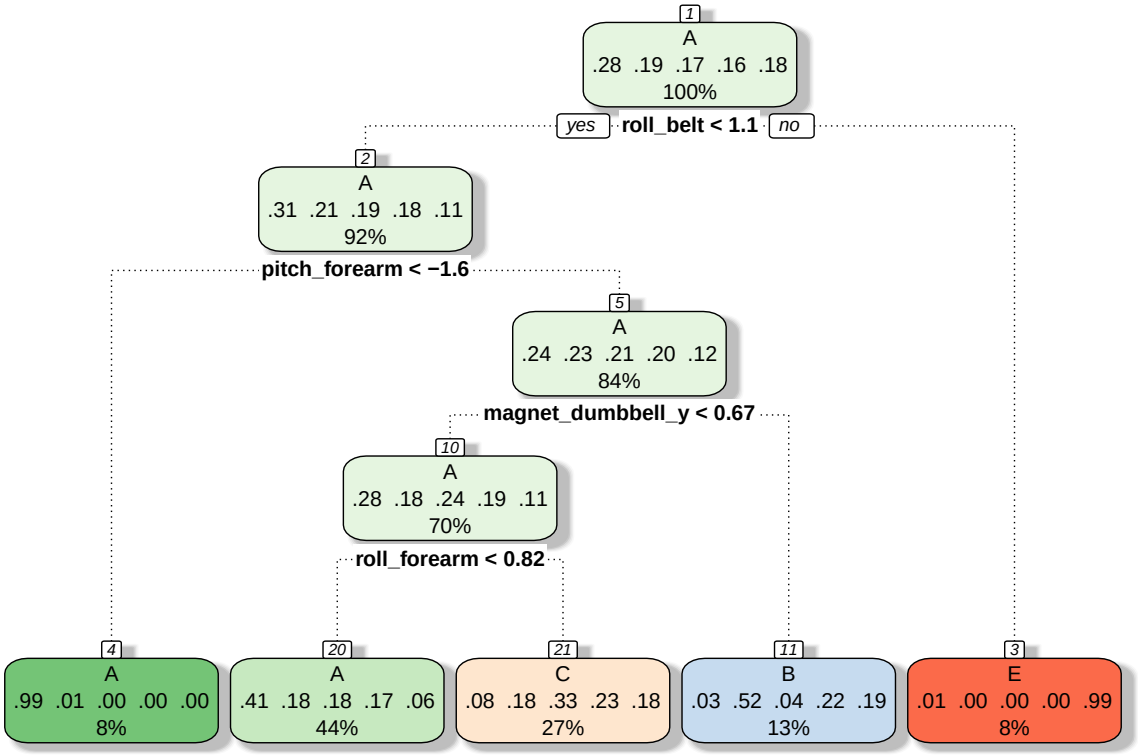
```
## CART
##
## 14718 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (53), scaled (53)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 14718, 14718, 14718, 14718, 14718, 14718, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.03840311  0.5497010  0.42302595
##   0.06101459  0.4075404  0.19392050
##   0.11535175  0.3297252  0.06924847
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03840311.
```

```
#Predict using our classification tree model
pred_rp <- predict(fit_rpart,testing)

#Use confusion matrix to check accuracy
confusionMatrix(pred_rp,testing$classe)$overall[[1]]
```

```
## [1] 0.4891925
```

```
#Plot the tree
fancyRpartPlot(fit_rpart$finalModel)
```

```
## Warning: Bad 'data' field in model 'call'.
## To silence this warning:
##     Call prp with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



Rattle 2018−Aug−11 22:10:15 andrewhu

** Random Forest **:

```
#randomForest
set.seed(777)
fit_rf <- train(classe ~ ., preProcess= c("center","scale"), method="rf",
                trControl=fitControl,verbose=FALSE,data=training)
#print our rf model
print(fit_rf)
```

```
## Random Forest
##
## 14718 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (53), scaled (53)
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 11774, 11775, 11774, 11773, 11776
## Resampling results across tuning parameters:
```

```
##
##   mtry  Accuracy   Kappa
##    2    0.9946323  0.9932100
##   27    0.9970105  0.9962186
##   53    0.9949039  0.9935535
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

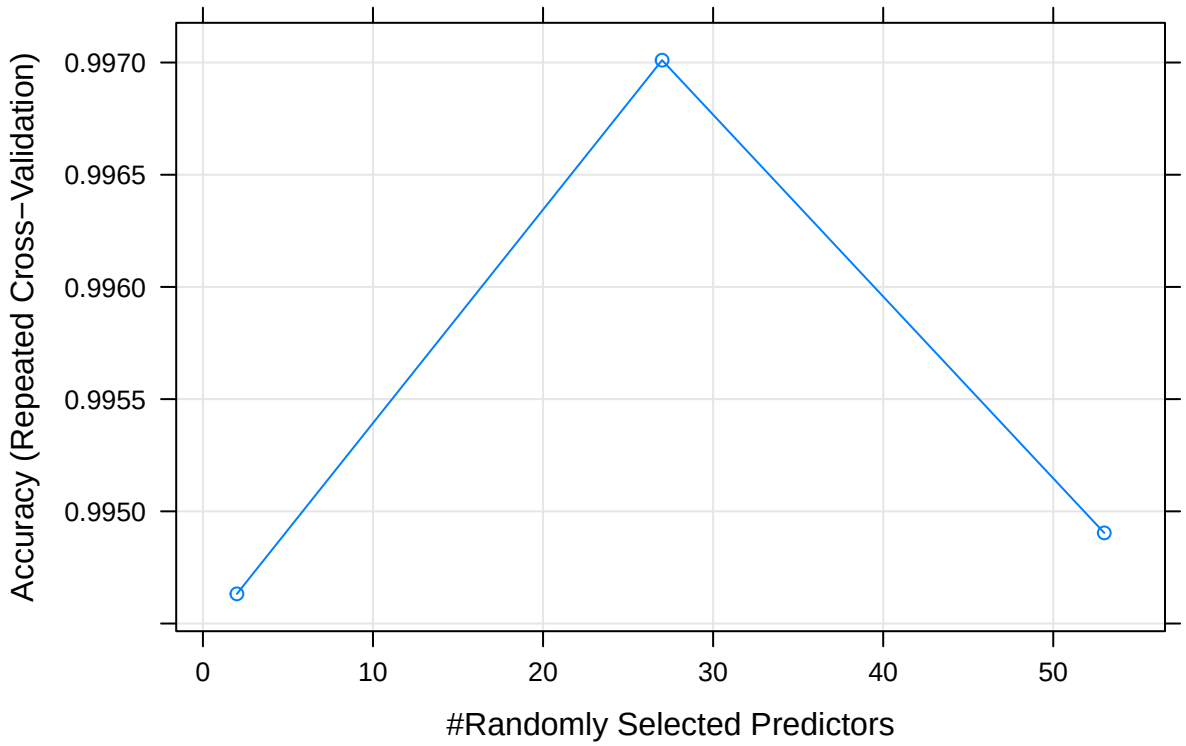
```
#Predict the testing data set using our rf model
pred_rf <- predict(fit_rf, testing)

#Use confusion matrix to check accuracy
confusionMatrix(pred_rf,testing$classe)$overall[[1]]
```

```
## [1] 0.9979608
```

```
#Check the importance of our variables
ImpVar <- varImp(fit_rf)
ImpVar
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 53)
##
##                      Overall
## num_window           100.000
## roll_belt             62.284
## pitch_forearm         39.453
## yaw_belt              29.801
## magnet_dumbbell_y     28.148
## magnet_dumbbell_z     27.957
## pitch_belt            27.439
## roll_forearm          22.768
## accel_dumbbell_y      13.063
## magnet_dumbbell_x     10.562
## roll_dumbbell          9.904
## accel_forearm_x        9.589
## accel_belt_z           8.890
## total_accel_dumbbell   8.629
## magnet_belt_y          7.844
## accel_dumbbell_z       7.429
## magnet_forearm_z       6.694
## magnet_belt_z          6.431
## magnet_belt_x          5.765
## accel_forearm_z        4.688
```

```
#Plot
plot(fit_rf, main="Accuracy of RF model by the number of predictors")
```



** lda model **:

```
#lda
set.seed(777)
fit_lda <- train(classe ~ ., preProcess= c("center","scale"), method="lda",
                 trControl=trainControl(method="cv", number=5),data=training)

pred_lda <- predict(fit_lda,testing)

confusionMatrix(pred_lda,testing$classe)$overall[[1]]
```

```
## [1] 0.7134992
```

** svm model **:

```
#svm
set.seed(777)
fit_svm <-svm(classe~., data=training)
pred_svm <- predict(fit_svm,testing)
confusionMatrix(pred_svm,testing$classe)$overall[[1]]
```

```
## [1] 0.9498369
```

** Gradient Boost Model **:

```
#gbm
set.seed(777)
fit_gbm <- train(classe~., method="gbm",preProcess= c("center","scale"),data=training,trControl=fitCont

print(fit_gbm)
```

```
## Stochastic Gradient Boosting
##
## 14718 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (53), scaled (53)
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 11774, 11775, 11774, 11773, 11776
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7569636  0.6914260
##   1                  100      0.8329926  0.7885721
##   1                  150      0.8690708  0.8342899
##   2                   50      0.8823210  0.8509570
##   2                  100      0.9389854  0.9227989
##   2                  150      0.9631737  0.9534107
##   3                   50      0.9319192  0.9138005
##   3                  100      0.9728219  0.9656137
##   3                  150      0.9871586  0.9837555
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
pred_gbm <- predict(fit_gbm,testing)

confusionMatrix(pred_gbm,testing$classe)$overall[[1]]
```

```
## [1] 0.9891925
```

** Combining model: randomForest, gradient boost model and lda **

```
comb_df <-data.frame(pred_gbm,pred_lda,pred_rf,classe=testing$classe)

#Stack all the models using random Forest
set.seed(777)
fit_comb <- train(classe~. , method="rf",preProcess= c("center","scale"),data=comb_df,trControl=fitCont

pred_comb <- predict(fit_comb,comb_df)
```

```r
confusionMatrix(pred_comb,testing$classe)$overall[[1]]
```

```
## [1] 0.9987765
```

Conclusion:

Comparing all the models above, I find that the combining model has the highest accuracy. Hence, the model combining with random Forest, gbm and lda is the most ideal one to predict what type of weight-lifting did the participants perfom in our study.