ML Course Project

```
# ------------------------------------------
#   data loading
# ------------------------------------------
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(33433)

# -- read data
train <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", head=TRUE, se
test <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", head=TRUE, sep=

# ------------------------------------------
#   data clean up
# ------------------------------------------
trainset <- createDataPartition(train$classe, p = 0.7, list = FALSE)
training <- train[trainset, ]
validation <- train[-trainset, ]

# -- exclude near zero features
nzvcol <- nearZeroVar(training)
training <- training[, -nzvcol]

# -- remove first 7 columns which don't contain useful info
training <- training[,-seq(1:7)]
dim(training)
```

```
## [1] 13737   121
```

```
# -- remove NAs
training <- training[, which(as.numeric(colSums(is.na(training)))==0)]
dim(training)
```

```
## [1] 13737    52
```

```r
# ------------------------------------------
#   model building
# ------------------------------------------
rfmod <- randomForest(classe ~ ., data = training, importance = TRUE)
ptraining <- predict(rfmod, training)
print(confusionMatrix(ptraining, training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```r
# ------------------------------------------
#   data validation
# ------------------------------------------
predValidation <- predict(rfmod, validation)
print(confusionMatrix(predValidation, validation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1671    5    0    0    0
##          B    3 1133    4    0    0
##          C    0    1 1022   17    0
##          D    0    0    0  945    4
```

```
##          E    0    0    0    2 1078
##
## Overall Statistics
##
##                Accuracy : 0.9939
##                  95% CI : (0.9915, 0.9957)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9923
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9947   0.9961   0.9803   0.9963
## Specificity            0.9988   0.9985   0.9963   0.9992   0.9996
## Pos Pred Value         0.9970   0.9939   0.9827   0.9958   0.9981
## Neg Pred Value         0.9993   0.9987   0.9992   0.9962   0.9992
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2839   0.1925   0.1737   0.1606   0.1832
## Detection Prevalence   0.2848   0.1937   0.1767   0.1613   0.1835
## Balanced Accuracy      0.9985   0.9966   0.9962   0.9897   0.9979
```

```r
# ------------------------------------------
#   data test
# ------------------------------------------
predTest <- predict(rfmod, test)
predTest
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```