Practial machine learning week4 assignment

The goal of this project

is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should 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.

load packages

```
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(caret)
```

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

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

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
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:ggplot2':
##
##     margin
```

load data

Data files were downloaded from the designated url.

```
train_url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
training_data <- read.csv(url(train_url))
```

data cleaning

Columns, which had index and many NA values (mean > 0.90), were removed. Moreover, colums with near zero-variation were removed.

```
# data cleaning; remove index column
training_data_01 <- training_data[,-c(1)]
# data cleaning; remove NAs
NA_col <- sapply(training_data_01, function(x) mean(is.na(x))) > 0.90
training_data_02 <- training_data_01[,NA_col == FALSE]
# data cleanining; non zero var
near_zero_var <- nearZeroVar(training_data_02)
training_data_03 <- training_data_02[,-near_zero_var]
```

data partition

60% of cleaned training data was used for building a model, and the remaining data was used for testing the model to evaluate its accuracy.

```
set.seed(191207)
inTrain <- createDataPartition(training_data_03$classe, p = 0.6, list = FALSE)
training <- training_data_03[inTrain,]
testing <- training_data_03[-inTrain,]
```

Random Forest model

Random Forest model was selected for the prediction model. And, cross-validation was used as a resampling method.

```
# parallel processing; n = 4
cl <- makePSOCKcluster(4)
registerDoParallel(cl)
# resampling method: cross validation
fitControl <- trainControl(method = "cv")
# building model
set.seed(0)
RF_mod <- train(classe ~.,
                  data = training,
                  method = "rf",
                  trControl=fitControl
                  )
# prediction and evaluation
RF_pred <- predict(RF_mod, testing)
RF_pred_conf <- confusionMatrix(RF_pred, testing$classe)
```

summary

In the case of test data, the model showed accuracy value of >0.99 and its sample error rate is acceptable. Based on this test result, further exploration of predictive models was not conducted.

```
RF_mod
```

```
## Random Forest 
## 
## 11776 samples
##    57 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E' 
## 
## No pre-processing
## Resampling: Cross-Validated (10 fold) 
## Summary of sample sizes: 10598, 10598, 10598, 10599, 10599, 10598, ... 
## Resampling results across tuning parameters:
## 
##   mtry  Accuracy   Kappa    
##    2    0.9886193  0.9856018
##   40    0.9980470  0.9975296
##   79    0.9971125  0.9963476
## 
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 40.
```

```
RF_pred_conf
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    4    0    0    0
##          B    0 1513    3    0    0
##          C    0    1 1360    3    0
##          D    0    0    5 1283    0
##          E    0    0    0    0 1442
## 
## Overall Statistics
##                                           
##                Accuracy : 0.998           
##                  95% CI : (0.9967, 0.9988)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9974          
##                                           
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9967   0.9942   0.9977   1.0000
## Specificity            0.9993   0.9995   0.9994   0.9992   1.0000
## Pos Pred Value         0.9982   0.9980   0.9971   0.9961   1.0000
## Neg Pred Value         1.0000   0.9992   0.9988   0.9995   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1928   0.1733   0.1635   0.1838
## Detection Prevalence   0.2850   0.1932   0.1738   0.1642   0.1838
## Balanced Accuracy      0.9996   0.9981   0.9968   0.9985   1.0000
```