# practical machine learning project

Yue Wang

7/7/2021

Building a machine learning model to predict the activity class

Load and read the training and testing data file

```
setwd("~/Desktop/coursera/Practical machine learning")
library(caret)
```

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

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

```
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
```

```
train<- read.csv("pml-training.csv")
test<- read.csv("pml-testing.csv")
```

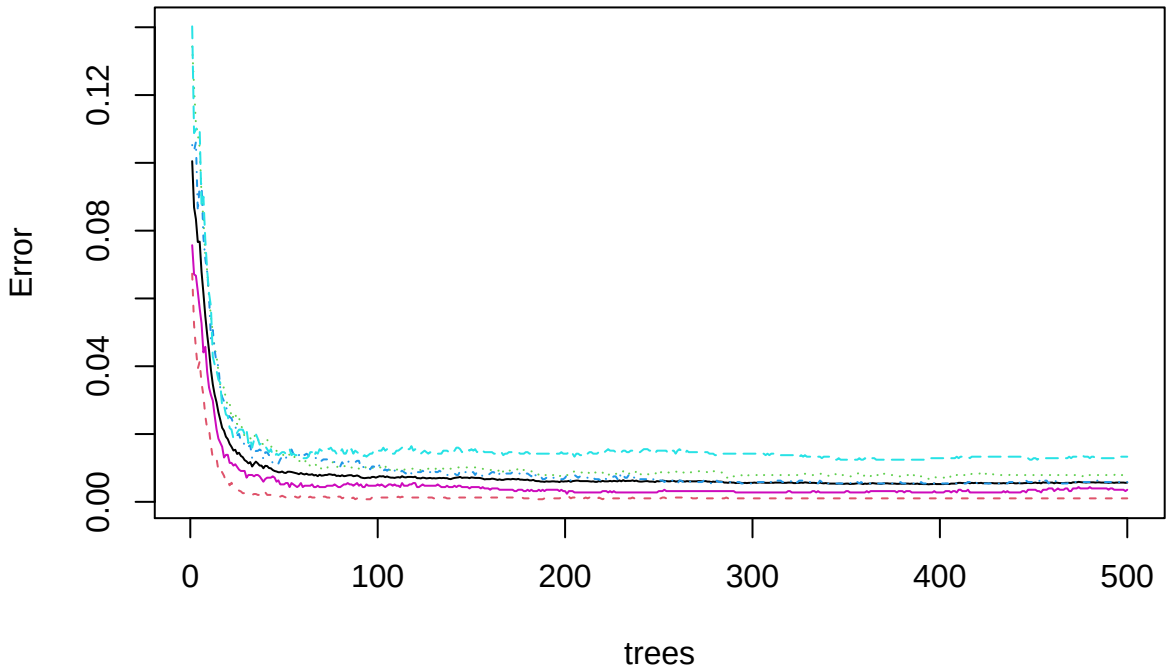
By reading the dataset and the original paper related to this research, we choose the following variables as predictors:

```
train1 <- subset(train, select=c(classe, roll_belt, pitch_belt, yaw_belt, total_accel_belt, gyros_belt_

test1 <- subset(test, select=c(roll_belt, pitch_belt, yaw_belt, total_accel_belt, gyros_belt_x, gyros_b
```

We use the random forest method to build the prediction model and perform k-fold cross validation (k=5):

```
set.seed(233)
train1$classe <- as.factor(train1$classe)
trainset <- createDataPartition(train1$classe, p = 0.7, list = FALSE)
trainsubtrain<- train1[trainset, ]
trainsubtest <- train1[-trainset, ]
train.control <- trainControl(method = "cv", number = 5)
fit1<- randomForest(classe~., data = trainsubtrain, trControl = train.control)
plot(fit1)
```




```
fit1
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = trainsubtrain, trControl = train.control)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.57%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3902    3    0    0    1 0.001024066
## B   15 2637    6    0    0 0.007900677
## C    0   10 2382    4    0 0.005843072
## D    0    0   27 2222    3 0.013321492
## E    0    0    2    7 2516 0.003564356
```

To show the expected out of sample error:

```
library(caret)
library(randomForest)
pred1<- predict(fit1, trainsubtest)
confusionMatrix(pred1, trainsubtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674   11    0    0    0
##          B    0 1125    7    0    0
```

```
##          C    0    3 1018    5    0
##          D    0    0    1  959    2
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.9951
##                  95% CI : (0.9929, 0.9967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9938
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9877   0.9922   0.9948   0.9982
## Specificity            0.9974   0.9985   0.9984   0.9994   1.0000
## Pos Pred Value         0.9935   0.9938   0.9922   0.9969   1.0000
## Neg Pred Value         1.0000   0.9971   0.9984   0.9990   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1912   0.1730   0.1630   0.1835
## Detection Prevalence   0.2863   0.1924   0.1743   0.1635   0.1835
## Balanced Accuracy      0.9987   0.9931   0.9953   0.9971   0.9991
```

We can see that this model has relative high accuracy in predicting activity classes. To predict the activity classes in the test1 dataset:

```
predict(fit1, test1)
```

```
##  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
```