Practical Machine Learning Prediction Assignment

Libraries

I used the following packages not listed in the course materials: randomForest package because the caret package run times were extremely long. http://cran.r-project.org/web/packages/randomForest/randomForest.pdf (http://cran.r-project.org/web/packages/randomForest/randomForest.pdf) I used the doParallel and foreach to speed up my http://cran.r-project.org/web/packages/doParallel/vignettes/gettingstartedParallel.pdf (http://cran.r-project.org/web/packages/doParallel/vignettes/gettingstartedParallel.pdf)

```
library(Hmisc)
library(caret)
library(randomForest)
library(foreach)
library(doParallel)
set.seed(998)
```

Loading Training Data

The pml-training.csv data is actually used to devise training and testing sets. The pml-test.csv data is used to predict and answer the 20 questions based on the trained model.

```
training.file   <- 'pml-training.csv'
test.cases.file <- 'pml-test.csv'
training.url    <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
test.cases.url  <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'

download.file(training.url, training.file)
download.file(test.cases.url,test.cases.file )
```

Cleaning Data

First all blank(“”’), ‘#DIV/0’ and ‘NA’ values are converted to ‘NA’. Any Columns containing ‘NA’ are removed from both downloaded data sets.

```
training.df   <-read.csv(training.file, na.strings=c("NA","#DIV/0!", ""))
test.cases.df <-read.csv(test.cases.file , na.strings=c("NA", "#DIV/0!", ""))
training.df<-training.df[,colSums(is.na(training.df)) == 0]
test.cases.df <-test.cases.df[,colSums(is.na(test.cases.df)) == 0]
```

The features user_name raw_timestamp_part_1 raw_timestamp_part_2 cvtd_timestamp new_window num_window are not related to calculations and are removed form the downloaded data.

```
training.df   <-training.df[,-c(1:7)]
test.cases.df <-test.cases.df[,-c(1:7)]
```

Create a stratified random sample of the data into training and test sets.seed(998)

```
inTraining.matrix    <- createDataPartition(training.df$classe, p = 0.75, list = FALSE)
training.data.df <- training.df[inTraining.matrix, ]
testing.data.df  <- training.df[-inTraining.matrix, ]
```

Use Random Forests

The outcome variable is ‘classe’ All variables other variables that assist in determining classe are defined as ‘variables’. The outcome variable is ’classe’are “fit’

```
registerDoParallel()
classe <- training.data.df$classe
variables <- training.data.df[-ncol(training.data.df)]
```

In the case of forest size we usemodel 1000 trees. We have four cores so we split up the problem into four pieces. This is accomplished by executing the randomForest function four times, with the ntree argument set to 250.[Using The foreach Package Steve Weston, doc@revolutionanalytics.com (mailto:doc@revolutionanalytics.com), April 10, 2014],[Package ‘randomForest’ July 17, 2014 Title Breiman and Cutler’s random forests for classification and regression Version 4.6-10 Date 2014-07-17]

```
rf <- foreach(ntree=rep(250, 4), .combine=randomForest::combine, .packages='randomForest') %d
opar% {
randomForest(variables, classe, ntree=ntree)
}
```

Confusion Matrix for Training

Predict and generate the Accuracy and confusion matrix for the training set (75% of the training data)

```
training.predictions <- predict(rf, newdata=training.data.df)
confusionMatrix(training.predictions,training.data.df$classe)
```

Confusion Matrix for Test set

Predict and generate the Accuracy and confusion matrix for the training set (25% of the testing data) Did the data overfit the training data?

```
testing.predictions <- predict(rf, newdata=testing.data.df)
confusionMatrix(testing.predictions,testing.data.df$classe)
```

Coursera provided code for submission

Set these `R` values from previous code that cleans data to match submission code.

```
feature_set <- colnames(training.df)
newdata     <- test.cases.df
```

Method to write answers to separate `.txt` files

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
```

Predict the `answers` to the 20 questions.

```
x <- evaluation_data
x <- x[feature_set[feature_set!='classe']]
answers <- predict(rf, newdata=x)
```

Now check

```
answers
```

Now write files and go to submission at coursera

```
pml_write_files(answers)
```