Course Project 1

Author R. Holla

Summary

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. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Loading and cleaning the data

First I load the necessary library and load the datasets. Because there are missing values, I replace them by NA.

```
library("caret")
```

```
## Warning: package 'caret' was built under R version 3.3.2
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library ("randomForest")
```

```
## Warning: package 'randomForest' was built under R version 3.3.3
```

```
## 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
```

```
TrainingSet<- read.csv("pml-training.csv", sep=",", header=TRUE, na.strings = c
("NA","",'#DIV/0!'))

ValidationSet<- read.csv("pml-testing.csv", sep=",", header=TRUE, na.strings =
c("NA","",'#DIV/0!'))
```

I split the Trainingset in a training part and a testing part.

```
set.seed(1000)

Train <- createDataPartition(y=TrainingSet$classe, p=0.75, list=FALSE)

Trainset <- TrainingSet[Train, ]

Testset <- TrainingSet[-Train, ]
```

Then we check which features aren’t zero in the ValidationSet. Then we know which columns we can use in the model.

```
UnUsefullColumns <- sapply(names(ValidationSet), function(x) all(is.na(Validati
onSet[,x])==TRUE))

UsefullColumns <- names(UnUsefullColumns)[UnUsefullColumns==FALSE]

UsefullColumns <- UsefullColumns[-(1:7)]

UsefullColumns <- UsefullColumns[1:(length(UsefullColumns)-1)]
```

Training the model

We make a random forest model and use crossvalidation.

```
RF <- train(classe ~.,
            method="rf",
            data=Trainset[, c('classe', UsefullColumns)],
            trControl=trainControl(method='cv'),
            number=5,
            allowParallel=TRUE
            )
```

Interpretation

Now we check the confusion matrix, the accuracy and the out-of-sample error of the model.

```
PRF <- predict(RF, Testset)

ConfMRF <- confusionMatrix(Testset$classe, PRF)

ConfMRF$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1392    3    0    0    0
##          B    4  944    1    0    0
##          C    0    5  838   12    0
##          D    0    0    5  796    3
##          E    0    0    0    6  895
```

```
Accuracy <- ConfMRF$overall[1]
Accuracy
```

```
##  Accuracy
## 0.9920473
```

```
OSE <- 1 - Accuracy
OSE
```

```
##    Accuracy
## 0.007952692
```

We can conclude that the Confusion Matrix looks very good, the accuracy is very high and the out-of-error sample is very low.

Prediction Quiz

Now we can make predictions for the quiz.

```
DefinitePredicitions <- predict(RF, ValidationSet)

DefinitePredicitions
```

```
##  [1] 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
```