Peer-graded Assignment: Prediction Assignment Writeup

Pavan

10/18/2020

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.3
```

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

```
## Warning: package 'lattice' was built under R version 3.6.3
```

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

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.6.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## 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
```

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.6.3
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

Getting the data

```
training <- read.csv("G://Notes//Sem - 7//Data Science//pml-training.csv")
testing <- read.csv("G://Notes//Sem - 7//Data Science//pml-testing.csv")
```

- Class A: exactly according to the specification
- Class B: elbows
- Class C: dumbbell lifting
- Class D: dumbbell lowering
- Class E: hips

Cleaning the data

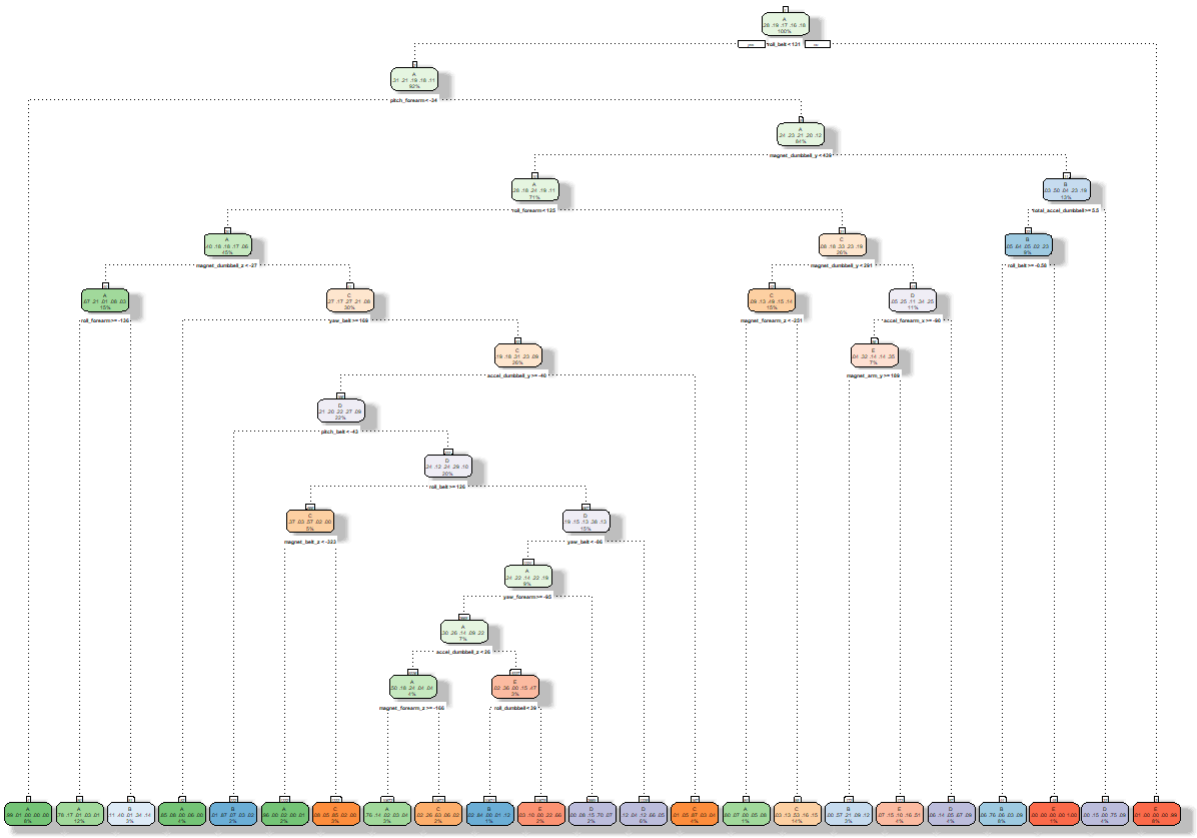
```
features <- names(testing[,colSums(is.na(testing)) == 0])[8:59]
trainproblem <- training[,c(features,"classe")]
testproblem <- testing[,c(features,"problem_id")]
```

Partitioning

```
inTrain <- createDataPartition(trainproblem$classe, p=0.7, list = FALSE)
TrainingCase <- trainproblem[inTrain,]
TestingCase <- trainproblem[-inTrain,]
```

Tree Prediction

```
DTmodel <- rpart(classe ~ ., data = TrainingCase, method = "class")
fancyRpartPlot(DTmodel)
```



Rattle 2020-Oct-18 17:26:39 MAHE

```
DTpredict <- predict(DTmodel, TestingCase, type = "class")
confusionMatrix(DTpredict, TestingCase$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1482  185   23   51   19
##          B   46  664   97   75   97
##          C   51  156  810  139  135
##          D   60   91   64  619   59
##          E   35   43   32   80  772
##
## Overall Statistics
##
##                Accuracy : 0.7387
##                  95% CI : (0.7272, 0.7498)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.669
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8853   0.5830   0.7895   0.6421   0.7135
## Specificity            0.9340   0.9336   0.9010   0.9443   0.9604
## Pos Pred Value         0.8420   0.6782   0.6274   0.6932   0.8025
## Neg Pred Value         0.9535   0.9032   0.9530   0.9309   0.9370
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2518   0.1128   0.1376   0.1052   0.1312
## Detection Prevalence   0.2991   0.1664   0.2194   0.1517   0.1635
## Balanced Accuracy      0.9096   0.7583   0.8452   0.7932   0.8370
```

Random Forest Prediction

```
RandomForestmodel <- randomForest(classe ~ ., data = TrainingCase)
RandomForestpredict <- predict(RandomForestmodel, TestingCase, type = "class")
confusionMatrix(RandomForestpredict, TestingCase$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1670    6    0    0    0
##          B    3 1131    9    0    0
##          C    0    2 1015   10    0
##          D    0    0    2  954    3
##          E    1    0    0    0 1079
##
## 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            0.9976   0.9930   0.9893   0.9896   0.9972
## Specificity            0.9986   0.9975   0.9975   0.9990   0.9998
## Pos Pred Value         0.9964   0.9895   0.9883   0.9948   0.9991
## Neg Pred Value         0.9990   0.9983   0.9977   0.9980   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2838   0.1922   0.1725   0.1621   0.1833
## Detection Prevalence   0.2848   0.1942   0.1745   0.1630   0.1835
## Balanced Accuracy      0.9981   0.9952   0.9934   0.9943   0.9985
```

Submission

In this, submission are generated using the random forest algorithm.

```
FinalPrediction <- predict(RandomForestmodel, testing, type = "class")
FinalPrediction
```

```
##  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
```

```
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)
    }
}
pml_write_files(FinalPrediction)
```