ml

Adnan

11/11/2020

Practical Machine Learning Project

Load library

```
## Warning: package 'caret' was built under R version 4.0.3
```

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

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

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
## Warning: package 'knitr' was built under R version 4.0.3
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## 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
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.3
```

Download and loading the Dataset

```
# Download the dataset
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
# Load the dataset into memory
trainingData <-  read.csv("./pml-training.csv",na.strings=c("NA","#DIV/0!",""))
testingData <- read.csv("./pml-testing.csv",na.strings=c("NA","#DIV/0!",""))
#
trainingData <- trainingData[, colSums(is.na(trainingData)) == 0]
testingData <- testingData[, colSums(is.na(testingData)) == 0]
# Delete variables that are not related
trainingData <- trainingData[, -c(1:7)]
testingData <- testingData[, -c(1:7)]
# partioning the training set into two different dataset
traningPartitionData <- createDataPartition(trainingData$classe,  p = 0.7, list = F)
trainingDataSet <- trainingData[traningPartitionData, ]
testingDataSet <- trainingData[-traningPartitionData, ]
dim(trainingData); dim(testingDataSet)
```

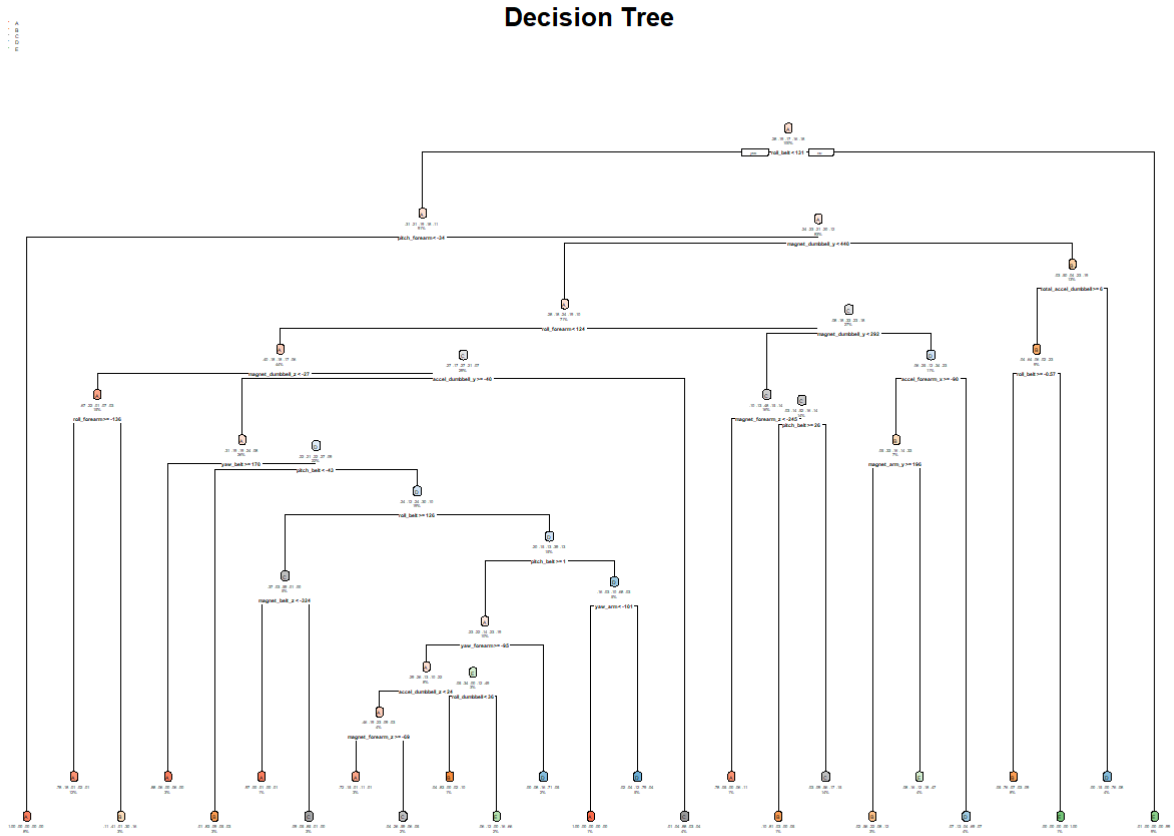
```
## [1] 19622    53
```

```
## [1] 5885   53
```

Prediction model 1 - decision tree

```
decisionTreeModel <- rpart(classe ~ ., data = trainingDataSet, method = "class")
decisionTreePrediction <- predict(decisionTreeModel, testingDataSet, type = "class")
# Plot Decision Tree
rpart.plot(decisionTreeModel, main = "Decision Tree", under = T, faclen = 0)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```




```
# Using confusion matrix to test results
confusionMatrix(factor(decisionTreePrediction, levels=1:10), factor(testingDataSet$classe, levels=1:10))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 1 2 3 4 5 6 7 8 9 10
##         1  0 0 0 0 0 0 0 0 0  0
##         2  0 0 0 0 0 0 0 0 0  0
##         3  0 0 0 0 0 0 0 0 0  0
##         4  0 0 0 0 0 0 0 0 0  0
##         5  0 0 0 0 0 0 0 0 0  0
##         6  0 0 0 0 0 0 0 0 0  0
##         7  0 0 0 0 0 0 0 0 0  0
##         8  0 0 0 0 0 0 0 0 0  0
##         9  0 0 0 0 0 0 0 0 0  0
##         10 0 0 0 0 0 0 0 0 0  0
##
## Overall Statistics
##
##                Accuracy : NaN
##                  95% CI : (NA, NA)
##     No Information Rate : NA
##     P-Value [Acc > NIR] : NA
##
##                   Kappa : NaN
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity               NA       NA       NA       NA       NA       NA
## Specificity               NA       NA       NA       NA       NA       NA
## Pos Pred Value            NA       NA       NA       NA       NA       NA
## Neg Pred Value            NA       NA       NA       NA       NA       NA
## Prevalence               NaN      NaN      NaN      NaN      NaN      NaN
## Detection Rate           NaN      NaN      NaN      NaN      NaN      NaN
## Detection Prevalence     NaN      NaN      NaN      NaN      NaN      NaN
## Balanced Accuracy         NA       NA       NA       NA       NA       NA
##                      Class: 7 Class: 8 Class: 9 Class: 10
## Sensitivity               NA       NA       NA        NA
```

```
## Specificity                NA       NA       NA       NA
## Pos Pred Value             NA       NA       NA       NA
## Neg Pred Value             NA       NA       NA       NA
## Prevalence                NaN      NaN      NaN      NaN
## Detection Rate            NaN      NaN      NaN      NaN
## Detection Prevalence      NaN      NaN      NaN      NaN
## Balanced Accuracy          NA       NA       NA       NA
```

Prediction model 2 - random forest

```
trainingDataSet$classe <- factor(trainingDataSet$classe)
randomForestModel <- randomForest(classe ~ ., data = trainingDataSet, method = "class")
randomForestPrediction <- predict(randomForestModel, testingDataSet, type = "class")
confusionMatrix(factor(randomForestPrediction, levels=1:10), factor(testingDataSet$classe, levels=1:10))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 1 2 3 4 5 6 7 8 9 10
##         1  0 0 0 0 0 0 0 0 0  0
##         2  0 0 0 0 0 0 0 0 0  0
##         3  0 0 0 0 0 0 0 0 0  0
##         4  0 0 0 0 0 0 0 0 0  0
##         5  0 0 0 0 0 0 0 0 0  0
##         6  0 0 0 0 0 0 0 0 0  0
##         7  0 0 0 0 0 0 0 0 0  0
##         8  0 0 0 0 0 0 0 0 0  0
##         9  0 0 0 0 0 0 0 0 0  0
##         10 0 0 0 0 0 0 0 0 0  0
##
## Overall Statistics
##
##                Accuracy : NaN
##                  95% CI : (NA, NA)
##     No Information Rate : NA
##     P-Value [Acc > NIR] : NA
##
##                   Kappa : NaN
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity               NA       NA       NA       NA       NA       NA
## Specificity               NA       NA       NA       NA       NA       NA
## Pos Pred Value            NA       NA       NA       NA       NA       NA
## Neg Pred Value            NA       NA       NA       NA       NA       NA
## Prevalence               NaN      NaN      NaN      NaN      NaN      NaN
## Detection Rate           NaN      NaN      NaN      NaN      NaN      NaN
## Detection Prevalence     NaN      NaN      NaN      NaN      NaN      NaN
## Balanced Accuracy         NA       NA       NA       NA       NA       NA
##                      Class: 7 Class: 8 Class: 9 Class: 10
## Sensitivity               NA       NA       NA        NA
```

```
## Specificity                 NA       NA       NA       NA
## Pos Pred Value              NA       NA       NA       NA
## Neg Pred Value              NA       NA       NA       NA
## Prevalence                 NaN      NaN      NaN      NaN
## Detection Rate             NaN      NaN      NaN      NaN
## Detection Prevalence       NaN      NaN      NaN      NaN
## Balanced Accuracy           NA       NA       NA       NA
```

Prediction model 2 - random forest

From the result, it show Random Forest accuracy is higher than Decision tree which is 0.9915 > 0.6644. Therefore, we will use random forest to answer the assignment.

```
predictionFinal <- predict(randomForestModel, testingDataSet, type = "class")
#predictionFinal
```