Prediction Assignment Writeup

Anjali Singh

16 October 2017

Background

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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

Data

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/ha (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/ha)

Training data https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

Test data https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Libraries

```
library(caret)
library(randomForest)
```

Loading Data

```
rm(list=ls())
tlink <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
vlink <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training <- read.csv(url(tlink))
validation<- read.csv(url(vlink))
```

Data Processing

Removing the near zero variance, NA values. We then removed the first 7 columns as they were not related to the classe variable. We also divided the data into Train and Test set

```
nzv <- nearZeroVar(training)
training<-training[,-nzv]

#removing predictors with NA values
training<-training[,colSums(is.na(training))==0]
training<-training[,-c(1:7)]
training$classe = factor(training$classe)
#Partition the data into Training and Testing
inTrain <- createDataPartition(y=training$classe, p=0.7, list=F)
training <- training[inTrain, ]
testing <- training[-inTrain, ]
```

Cross Validation

We will be doing 3 fold cross validation to train our model using first Rtree method and then random forest method later. The one with better accuracy will be selected the right model technique.

```
set.seed(9999)
cv = trainControl(method="cv",number=5,allowParallel=TRUE,verboseIter=TRUE)
modelrf = train(classe~., data=training, method="rf",trControl=cv)#Randon forest
modeltree = train(classe~.,data=training,method="rpart",trControl=cv)# RTree
```

Checking for accurancy on training and testing data. For random forest

```
#training data
prediction_rf1 <- predict(modelrf,newdata=training)
confusionMatrix(prediction_rf1,training$classe)
```

```
#testing data
prediction_rf2 <- predict(modelrf,newdata=testing)
confusionMatrix(prediction_rf2,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1182    3    0    0    0
##          B    1  769    0    0    0
##          C    0    0  750    2    0
##          D    0    0    0  657    0
##          E    0    0    0    0  765
##
## Overall Statistics
##
##                Accuracy : 0.9985
##                  95% CI : (0.9968, 0.9995)
##     No Information Rate : 0.2865
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9982
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9992   0.9961   1.0000   0.9970   1.0000
## Specificity            0.9990   0.9997   0.9994   1.0000   1.0000
## Pos Pred Value         0.9975   0.9987   0.9973   1.0000   1.0000
## Neg Pred Value         0.9997   0.9991   1.0000   0.9994   1.0000
## Prevalence             0.2865   0.1870   0.1816   0.1596   0.1853
## Detection Rate         0.2863   0.1862   0.1816   0.1591   0.1853
## Detection Prevalence   0.2870   0.1865   0.1821   0.1591   0.1853
## Balanced Accuracy      0.9991   0.9979   0.9997   0.9985   1.0000
```

For RTree

```
#training data
prediction_tree1 <- predict(modeltree,newdata=training)
confusionMatrix(prediction_tree1,training$classe)
```

```
#testing data
prediction_tree2 <- predict(modeltree,newdata=testing)
confusionMatrix(prediction_tree2,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1052  307  332  306  185
##          B   23  271   22  124  155
##          C   77   80  342   80  113
##          D   29  114   54  149  108
##          E    2    0    0    0  204
##
## Overall Statistics
##
##                Accuracy : 0.4887
##                  95% CI : (0.4734, 0.5041)
##     No Information Rate : 0.2865
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3312
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8893  0.35104  0.45600  0.22610  0.26667
## Specificity            0.6164  0.90349  0.89642  0.91210  0.99941
## Pos Pred Value         0.4821  0.45546  0.49422  0.32819  0.99029
## Neg Pred Value         0.9327  0.85823  0.88129  0.86122  0.85700
## Prevalence             0.2865  0.18697  0.18164  0.15960  0.18527
## Detection Rate         0.2548  0.06563  0.08283  0.03609  0.04941
## Detection Prevalence   0.5285  0.14410  0.16760  0.10995  0.04989
## Balanced Accuracy      0.7528  0.62726  0.67621  0.56910  0.63304
```

The accuracy for Rtree(.49)is less than Random Forest(0.99). We will use Random forest on validation data

Validation

```
nzv <- nearZeroVar(validation)
validation<-validation[,-nzv]

#removing predictors with NA values
validation<-validation[,colSums(is.na(validation))==0]
validation<-validation[,-c(1:7)]

predictionFinal<- predict(modelrf, newdata=validation)
predictionFinal
```

```
##  [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
```

Conclusion

We got all the prediction right