Practical Machine Learning - Prediction Assignment

Ronald

9 Jan 2021

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, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 young healthy participants who were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions:

i. exactly according to the specification (Class A);
ii. throwing the elbows to the front (Class B);
iii. lifting the dumbbell only halfway (Class C);
iv. lowering the dumbbell only halfway (Class D); and
v. throwing the hips to the front (Class E).

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

The training data for this project are available here:

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

The test data are available here:

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

Load Data

```
# Set Seed
set.seed(2)

# Load Libraries
library(tidyverse)
library(caret)
library(rattle)

# Load data
train.data <- read.csv(file = "pml-training.csv", na.strings=c("NA","#DIV/0!",""))
test.data <- read.csv(file = "pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Data Cleaning

```
#Remove Unnecessary columns
train.data <- train.data [-c(1:7)]
test.data <- test.data [-c(1:7)]

#Remove NA columns
train.data.clean <- train.data [colSums(is.na(train.data)) == 0]
test.data.clean <- test.data [colSums(is.na(train.data)) == 0]
```

Classification Models

Classification models were built using the training data set. Decision Tree model was first used to try to classify the data.

To allow the use of all training data set for model building, 10-fold cross validation technique was used to evaluate the model. Cross-validation partition the original sample into a training set to train the model and a test set to evaluate it.

Decision Tree

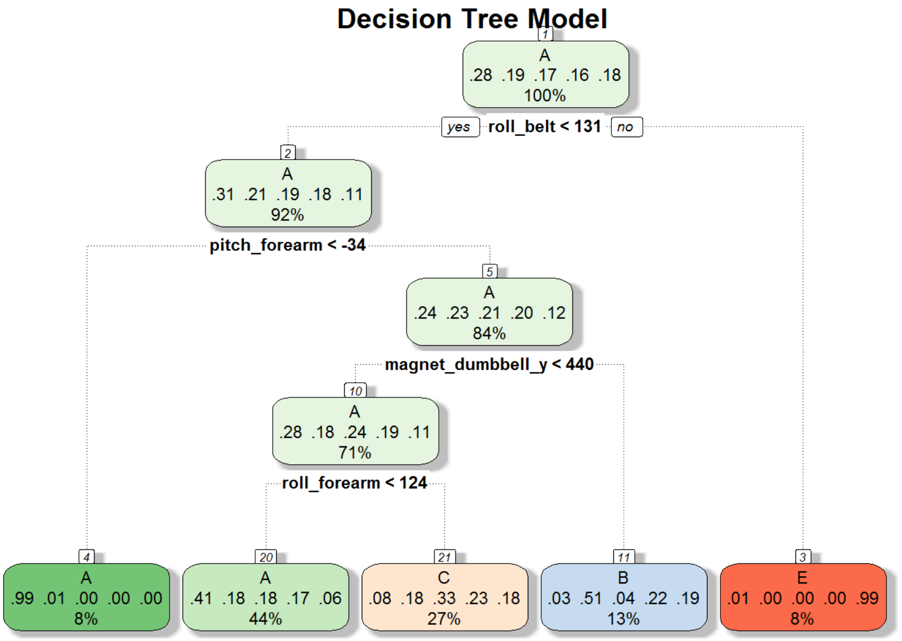
```
control.par <- trainControl(method="cv", 10)

Tree_model <- train(classe ~ ., data=train.data.clean, method="rpart", trControl=control.par)

Tree_model
```

```
## CART
##
## 19622 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17661, 17659, 17659, 17660, 17659, 17659, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.03567868  0.5065707  0.35552444
##   0.05998671  0.4156034  0.20827406
##   0.11515454  0.3246439  0.06137408
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03567868.
```

```
fancyRpartPlot(Tree_model$finalModel, main = "Decision Tree Model\n", sub = "")
```



```
confusionMatrix(Tree_model)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 25.9  8.1  8.1  7.4  2.7
##          B  0.4  6.6  0.6  2.9  2.5
##          C  1.8  3.3  8.1  4.3  3.5
##          D  0.3  1.4  0.7  1.8  1.4
##          E  0.1  0.0  0.0  0.0  8.3
##
##  Accuracy (average) : 0.5066
```

The accuracy of the decision tree model was only **0.5066**. With a low accuracy, the decision tree model was not suitable to be used as a prediction model.

To increase the accuracy, Random forest technique was then used. Random forest technique is an ensemble learning method for classification that operate by constructing a multiple decision trees. The technique often yield better classification results than the single decision tree model.

```
Rf_model <- train(classe ~ ., data=train.data.clean, method="rf",  trControl=control.par, ntree=100)

Rf_model
```

```
## Random Forest
##
## 19622 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17659, 17659, 17660, 17660, 17660, 17660, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9950055  0.9936822
##   27    0.9950565  0.9937465
##   52    0.9899093  0.9872346
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
confusionMatrix(Rf_model)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.1  0.0  0.0  0.0
##          B  0.0 19.2  0.1  0.0  0.0
##          C  0.0  0.0 17.3  0.1  0.0
##          D  0.0  0.0  0.1 16.2  0.0
##          E  0.0  0.0  0.0  0.0 18.3
##
##  Accuracy (average) : 0.9951
```

The accuracy of the Random forest model was **0.9951**. The Random forest model is therefore fairly accurate.

Prediction

Using the Random forest model built, prediction of the outcomes of the test data set was performed.

```
classificaion.result <- predict(Rf_model, newdata = test.data.clean)

classificaion.result
```

```
##  [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
```