Practical Machine Learning: Weightlifting Prediction Exercise

Shivam Giri

September 17, 2017

Overview

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.In this project, Our goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and to predict the manner in which they did the exercise. More information is available from the website (http://groupware.les.inf.puc-rio.br/har)

Loading the Data

```r
train_raw<-read.csv("F:/Downloads/pml-training.csv",header = TRUE)
validation<-read.csv("F:/Downloads/pml-testing.csv",header = TRUE)
```

Cleaning the Data

```r
#Some coloumn contains NAs in excess
maxNA_perc<-20
maxNA_count<-nrow(train_raw)/100*maxNA_perc

#Insignificant Columns(NAs) & time series columns
insig_column<-which(colSums(is.na(train_raw)|train_raw=="")>maxNA_count)
ts_col<-grep("timestamp",names(train_raw))

#Removing Time series Columns and insignificant columns
train_cleaned<-train_raw[,-c(1,ts_col,insig_column)]
validation_cleaned<-validation[,-c(1,ts_col,insig_column)]
```

Data Slicing

```r
library(caret)
```

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

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

```r
library(ggplot2)
library(lattice)
set.seed(2334)
inTrain<-createDataPartition(train_cleaned$classe,p=.7,list = FALSE)
training<-train_cleaned[inTrain,]
testing<-train_cleaned[-inTrain,]
```

Exploratory data analysis

```r
dim(training)
dim(testing)
str(training)
str(testing)
```

Model Selection

For this project I'll use 3 differnt model algorithms and then look to see which provides the best out-of-sample accuracty. The three model types I'm going to test are:

1)Decision trees with CART (rpart) 2)Stochastic gradient boosting trees (gbm) 3)Random forest decision trees (rf)

```r
library(survival)
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```r
model_rpart<-train(classe~.,data = training,method="rpart")
model_gbm<-train(classe~.,data = training,method="gbm",verbose=FALSE)
```

```
## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

```r
model_rf<-train(classe~.,data = training,method="rf")
```

```
## 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
```

CART(rpart) model

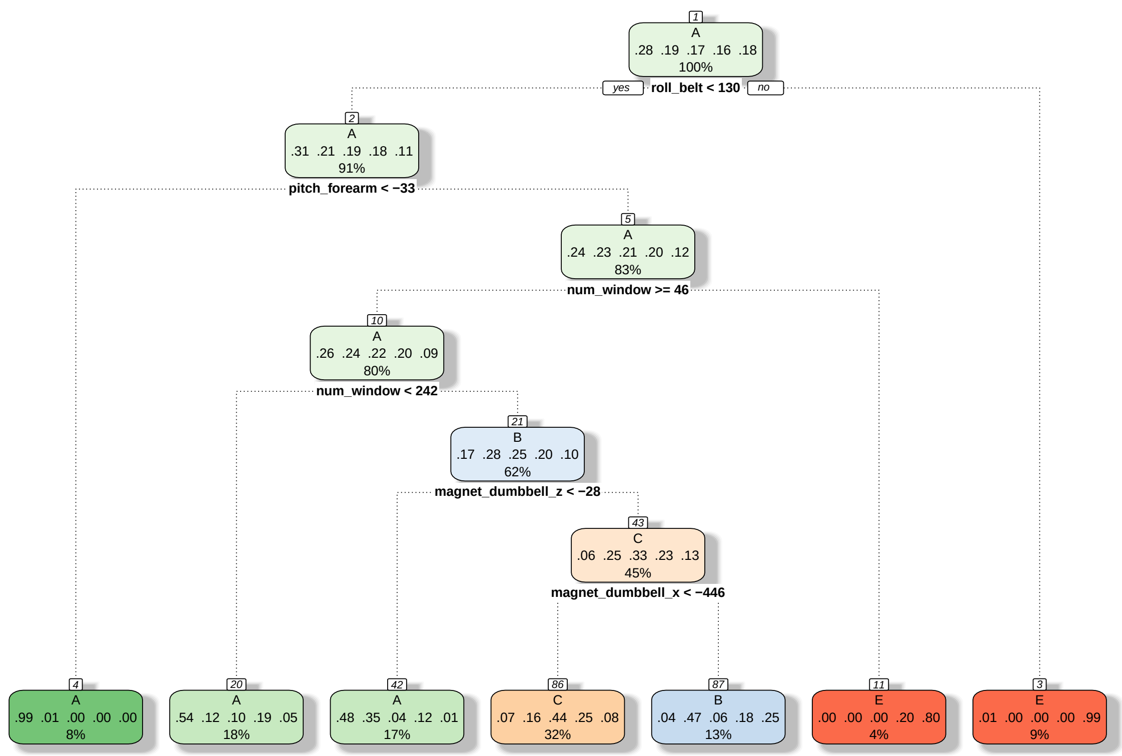
```r
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

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

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

```r
fancyRpartPlot(model_rpart$finalModel)
```



Rattle 2017−Sep−17 22:14:26 dell

Model Assessment

```r
#Predictions on testing dataset
pred_rpart<-predict(model_rpart,newdata = testing)
pred_gbm<-predict(model_gbm,newdata = testing)
pred_rf<-predict(model_rf,newdata = testing)

#Confusion matrix(Accuracy) of different models
cm_rpart<-confusionMatrix(testing$classe,pred_rpart)$overall
cm_gbm<-confusionMatrix(testing$classe,pred_gbm)$overall
cm_rf<-confusionMatrix(testing$classe,pred_rf)$overall
data.frame(Model=c("Cart","gbm","rf"),Accuracy=c(cm_rpart[1],cm_gbm[1],cm_rf[1]))
```

```
##   Model  Accuracy
## 1  Cart 0.5559898
## 2   gbm 0.9860663
## 3    rf 0.9983008
```

The next step should be to create an ensemble model, but given the very high accuracy of ‘rf models, we will adopt ’rf’ model as the final model.

Prediction

Performing prediction on validation dataset as the final step to evaluate the model

```r
pred_valid<-predict(model_rf,newdata = validation_cleaned)
final_pred<-data.frame("Problem_id"=validation_cleaned$problem_id,"Predicted Value"=pred_valid)
print(final_pred)
```

```
##    Problem_id Predicted.Value
## 1           1               B
## 2           2               A
## 3           3               B
## 4           4               A
## 5           5               A
## 6           6               E
## 7           7               D
## 8           8               B
## 9           9               A
## 10         10               A
## 11         11               B
## 12         12               C
## 13         13               B
## 14         14               A
## 15         15               E
## 16         16               E
## 17         17               A
## 18         18               B
## 19         19               B
## 20         20               B
```

Conclusion

The random forest model with cross-validation produces a surprisingly accurate model with 99.83% accuracy that is sufficient for predictive analytics.