Machine_learning_assignment

Anusha Iyengar

Wednesday, April 20, 2016

Introduction

This is a report of the predictive analysis done on fitness data collected for 6 different users. The goal of the analysis is to predict *how* someone is going to perform based on the data collected about their various work out movements.

Problem Description

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://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

The test data are available here:

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

Predictive Analysis

My approach to going about with the analysis was the following. a) Analyse dataset to determine columns that werent' adding value to the response variable and thereby removing them b) Further with the clean dataset, split it into a Training and Test dataset c) Start with the RPART predictive modeling method and check accuracy of the prediction d) If not satisfactory then run the Random Forest method. e) If accuracy is upto 99% stop further analysis, if not - continue with other methods.

Loading Data

```r
fit.train <- read.csv(file="pml-training.csv", na.strings = c("NA",""),header=TRUE)
fit.test <- read.csv(file="pml-testing.csv", header=TRUE)
```

Observe the dataset

```r
str(fit.train)
```

There are a total of 19622 observations and 160 variables. There seem to be several variables that with a number of NA values. Let us look at the summary function to find out the number of NAs in the columns

```r
summary(fit.train )
```

It appears as though the columms with NAs have a total of 19216 NAs out of a total of 19622. That is almost all the values in the column. Let's eliminate these columns with high number of NA's, as they do not add value to our data analysis and will only distort the results.

```r
fit.train.2 <- fit.train[!colSums(is.na(fit.train)) > 19215]
str(fit.train.2)
```

A quick and easy way to count the number of NAs by column in the fit.train dataset.

```r
colSums(is.na(fit.train.2))
```

It is evident that the remaining columns do not have a high number of NAs

Further excluding variables “X”,“user_name”,raw_timestamp_part_1“,”raw_timestamp_part_2“,”cvtd_timestamp“,”new_w
from the dataset as these don't add any value to the response variable *classe*

```r
remove.vars <- names(fit.train.2) %in% c("X","user_name","raw_timestamp_part_1","raw_timestamp_part_2",
fit.train.final <- fit.train.2[!remove.vars]
```

Now to verify if any of the columns have values with zero or near zero variance

```r
nearZero <- nearZeroVar(fit.train.final, saveMetrics=TRUE)
nearZero
```

It is evident that none of the columns have zero or near zero variance, so there isn't any further need to reduce columns from the dataset.

Now that the dataset is clean, let's move on to carrying out some predictive analysis on it. First, since the dataset has over 19000 records, let's split the dataset into Training and test datasets. We'll move 75% of the data into the training set and remaining 25% in a test set.

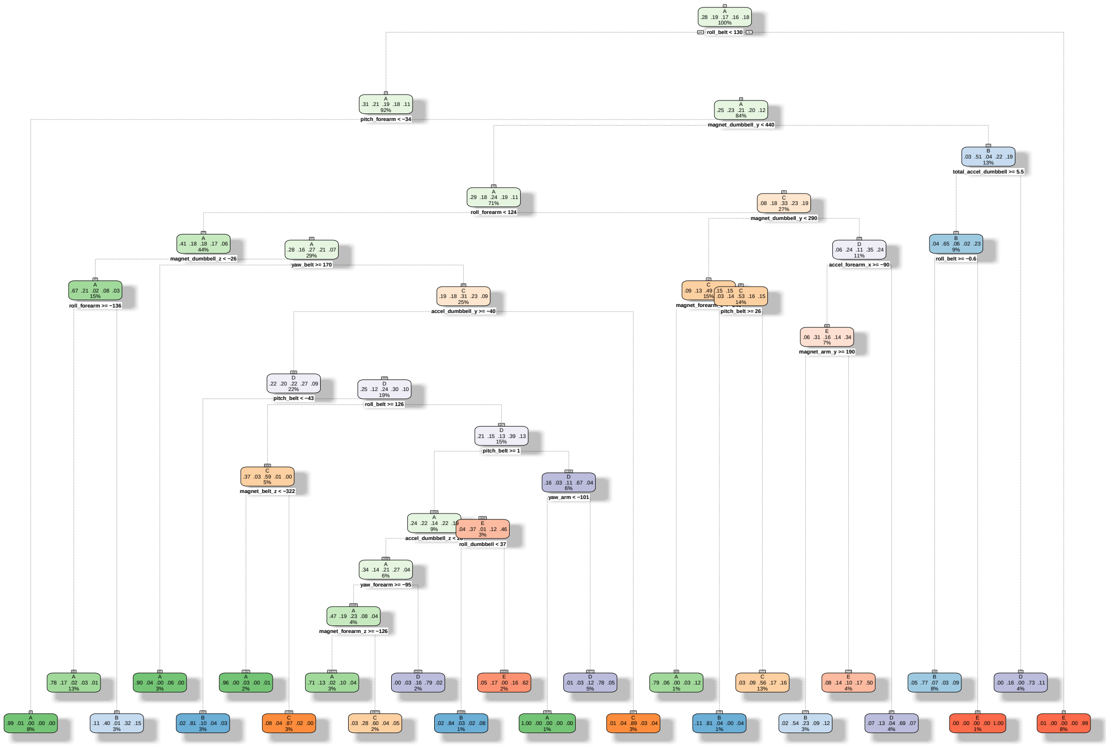
```r
inTrain <- createDataPartition(y=fit.train.final$classe, p = 0.75, list=FALSE)
training <- fit.train.final[inTrain,]
testing <- fit.train.final[-inTrain,]

dim(training)
dim(testing)
```

Predictive analysis on dataset training

```r
set.seed(1234)
modFit <- rpart(classe ~ ., data=training, method="class")
print(modFit)
```

```r
fancyRpartPlot(modFit)
```

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



Rattle 2016−Apr−24 21:14:31 Iyengar

```r
preFit <- predict(modFit,testing, type="class")
```

Let's observe the accuracy and statistics by using the confusionMatrix function

```r
confusionMatrix(preFit, testing$classe)
```

An accuracy of 74% is not that great!

Now let's try the random forest method in hopes of getting a higher prediction accuracy.

```r
modFit.rf <- randomForest(classe ~ ., data=training)
```

Now lets run prediction for our Testing data set.

```r
pred.rf <- predict(modFit.rf, testing, type ="class")
```

ConfusionMatrix to view Accuracy of the prediction.

```
confusionMatrix(pred.rf, testing$classe)
```

The Random Forest method has yielded a very good accuracy of 0.99 ie 99%

Final test with 20 test cases.

Thus the model predicted by the Random Forest method is a good one and I will use it to further predict the outcomes of the 20 test cases fit.test is the dataset that contains the 20 test cases.

```
pref.test <- predict(modFit.rf, fit.test, type = "class")
pref.test
```

```
##  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
```