Weight Lifting Exercise Prediction

Ana Baraldi

February 26, 2016

Summary

This project's goal is to predict which way each user is doing Weight Lifting Exercise in the test set. We have five ways (A, B, C, D, E) to do the exercise. I've chosen Tree Prediction to train our model because it is the easiest way to understand the outcome, but we didn't get a good accuracy in the model. Then I've chosen a Random Forest model and now the accuracy was great!

First of all we are going to load the libraries:

```
library(dplyr)
library(caret)
library(rpart)
library(randomForest)
library(rattle)
```

Reading Data

```
if (!file.exists("pml-training.csv")) {
    download.file(url ="https://d396qusza40orc.cloudfront.net/predmachlearn/pml
-training.csv", destfile = "pml-training.csv")
}
pml_training <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""), h
eader=TRUE)

if (!file.exists("pml-testing.csv")) {
    download.file(url ="https://d396qusza40orc.cloudfront.net/predmachlearn/pml
-testing.csv", destfile = "pml-testing.csv")
}
pml_testing <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""), hea
der=TRUE)
```

Our data is already splitted into training and testing, just to have an idea, our training set has 19622 observations and 160 different variable measured. Our testing is way smaller and has only 20 observations.

Cleaning Data

While I was trying to build the model, I've had some problems with some measured variables that had a lot of `NA` values, that's why I've decided to remove those variables. I've also removed the first columns that were only control variables as `user_name`, `X` and a few others. I've created a function `clean_data` to run those cleaning procedures.

```
clean_data <- function(dataset) {
    cleaned <- dataset %>% dplyr::select(-c(X, user_name, raw_timestamp_part_1,
raw_timestamp_part_2, cvtd_timestamp, new_window, num_window))
    cleaned <- cleaned[sapply(cleaned, function(x) !any(is.na(x)))]
    return(cleaned)
}

cleaned_train <- clean_data(pml_training)
```

Create Training and Validating

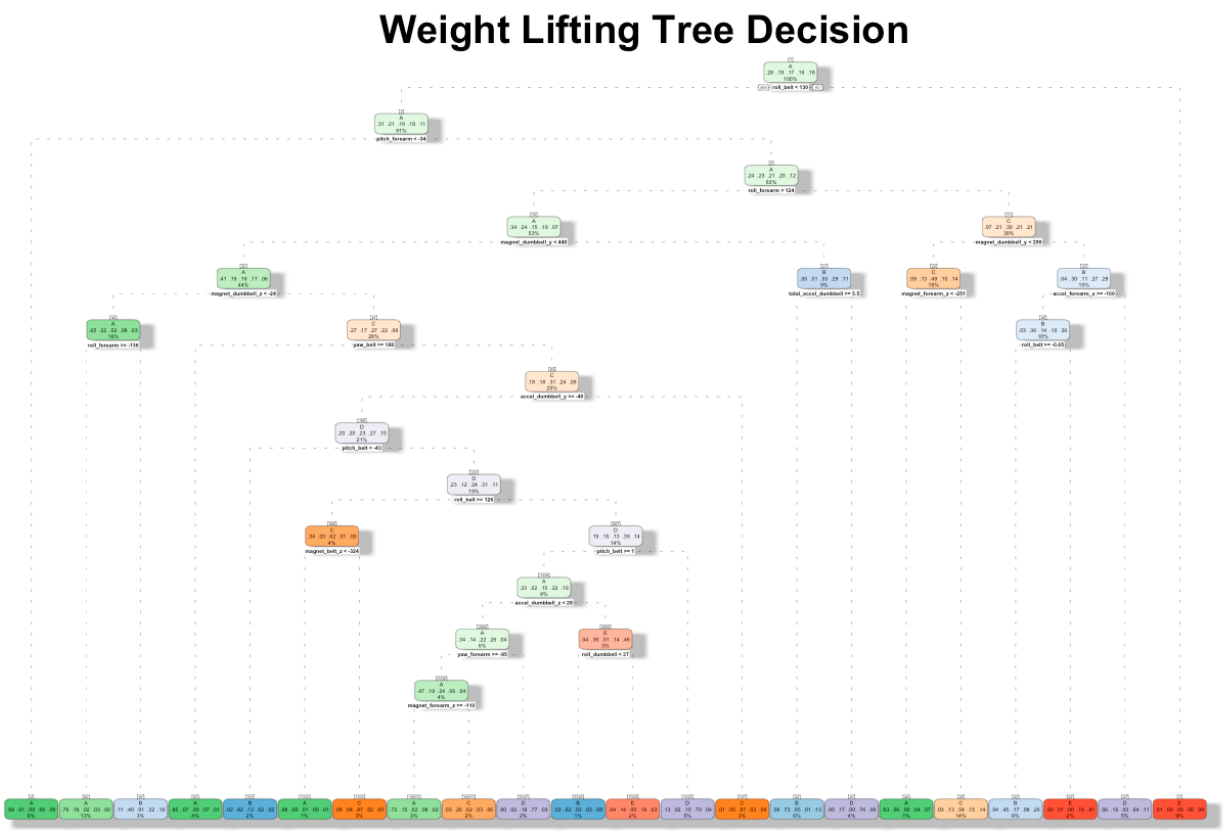
As we have a big dataset with 19622 observations, I've decided to split my data one more time and validate my model before I use it in the test set.

```
inTrain <- createDataPartition(y=cleaned_train$classe, p=0.70, list=FALSE)
training <- cleaned_train[inTrain, ]
validating <- cleaned_train[-inTrain, ]
```

Create Tree Model

Here we are going to create our model and check some information about it, we are going to print our Tree Decision as well.

```
fit <- rpart(classe ~ ., data=training, method="class")
fancyRpartPlot(fit, sub=NULL, main= "Weight Lifting Tree Decision")
```



Validate our model

Here we are going to validate our model usint the `predict` function and we are going to print our confusion matrix to see how we've performed in our validating set.

```
pred <- predict(fit, newdata=validating, type="class")
accuracy <- postResample(pred, validating$classe)[1]
```

As we can see our accuracy was 0.7308411. As we need at least 80% of accuracy we are going to try another model, I've chose Random Forest to continue our study because it uses Tree Decisions as part of the model.

Create Random Forest

```
fit2 <- randomForest(classe ~ ., data=training)
pred2 <- predict(fit2, newdata=validating)
accuracy2 <- postResample(pred2, validating$classe)[1]
```

Now our accuracy is 0.9943925, which is a lot better. That's why we are going to choose this model as our final model! :)

Predict test set

Here are my predictions for the test set!

```
predict(fit2, newdata=pml_testing[names(pml_testing) %in% names(training)])
```

```
##  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
```