Machine Learning PDF

Samantha Stabley

November 26, 2017

```
knitr::opts_chunk$set(echo = TRUE)
```

Executive Summary

In this project I am interested in exploring the relationship between a set of variables and if an exercise is being performed correctly. My main goal being, that I am able to create a prediction model that can use imput from various devices such as Jawbone Up, Nike FuelBand, and Fitbit to predict whether a participant is correctly performing barbell lifts.

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

Note that the "Classe" variable states whether a participant is correctly performing a barbell lift or if they are making one of the mistakes predetermined.

This analysis will show that the random forest function in R provides a very good predictor for the information given.

Exploratory Data Analysis

```
#Dataset:
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
library(rpart)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.3.3

## 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
```

```
DataOrig=read.csv("C:/Users/Samantha/Desktop/data_science/Machine Learning/pml-training.csv", na.string

#For reproducibility
set.seed(222)
```

Some of the data columns are not pertinent to our needs. For instance, a participants name or time of day should not be taken into account of whether they are correctly doing the lift. Hence, I got rid of columns 1-7 for these reasons.

Also, many columnshave LOTS of NAs, for some reason many columns had exactly 19216 NAs out of 19622 observations (about 97.9% of the column is NA or blank) which causes many errors and offers little information. So I also deleted any column with this many NAs in it.

```
data<-DataOrig[,-c(1:7)]
Sum<-colSums(is.na(data))
true<-Sum==0
numbers=vector()
x=1
for (i in 1:153){
    if (Sum[i]==19216){
    numbers[x]<-i
    x<-x+1

} }
data<-data[,-numbers]
```

In order to do cross validation I need to split my training set into training and validation sections so I can perform an accurate out of sample error rate. I used createDataPartition to do this.

Cross Validation

```
data1<- createDataPartition(y=data$classe, p=0.7, list=FALSE)
val<- data[-data1,]
training<-data[data1,]
```

Choosing models to accurately describe the data was tricky, I looked at several quick plot layouts before attempting rpart, LDA, and random forest.

Fitting models based on training data

```
mod_rpart<-rpart(classe~., method="class", data=training)
mod_lda<-train(classe~., method="lda", data=training)
mod_rf<- randomForest(classe ~. , data=training)
```

From these I made predictions and tested them using confusionMatrix()

```
pred_rpart<-predict(mod_rpart, newdata= val)
pred_lda<-predict(mod_lda, newdata = val)
pred_rf<-predict(mod_rf, newdata = val)
confusionMatrix(pred_lda,val$classe)$overall[1]
```

```
##  Accuracy
## 0.7019541
```

```r
confusionMatrix(pred_rf,val$classe)$overall[1]
```

```
##  Accuracy 
## 0.9954121
```

Why I chose Random Forest method

I used my validation test set (val) to test my models, and come up with accuracy rates. I was originally planning on stacking my models, but the random forest method had a 99% accuracy rate, so I didn't feel stacking could improve on this.

Testing my Model

```r
testing=read.csv("C:/Users/Samantha/Desktop/data_science/Machine Learning/pml-testing.csv", na.strings =

#I need to remove the exact same columns from my test set that I removed from my training set
testing<-testing[,-c(1:7)]
testing<-testing[,-numbers]
test_pred<-predict(mod_rf,newdata= testing)
test_pred
```

```
##  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
```

Now that I have my test prediction I need to calculate my out of sample error. I will use the validation set to do this and calculate the RMSE (root mean square error), which totals 27 for my study.

Out of Sample Error

```r
predicted<-predict(mod_rf,newdata = val)
actual<- val$classe
Correct<-actual==predicted
Difference<- length(actual) - sum(Correct)
sqrt(mean((Difference)^2))
```

```
## [1] 27
```