ML_Project

Adolfo Morales

May 26, 2017

Assignment

The document below details a Machine Learning approach to attempt to identify whether certain exercises are performed correctly. The data is gathered from Fitness Trackers worn by the subjects of the experiment. The data should be classified into 5 categories, Category A representing a correctly executed exercise, Categories B - E each describe a particular error.

Getting Data

The data was downloaded from the assignment website, and is being read in the section below. Required Libraries are also read into memory.

```
setwd("C:/Users/amora/OneDrive/Documents/R_Projects/Coursera/Machine_Learning/Project")
training <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""), header=TRUE)
testing <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""), header=TRUE)

set.seed(10)
library(lattice)
```

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

```
library(ggplot2)
library(caret)
```

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

```
library(randomForest)
```

```
## 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
```

Splitting Training into Training/Validation

In order to test the model to be fitted, the training data will be partitioned into a Training and a Validation set.

```
inTrain <- createDataPartition(training$classe, p=0.7, list=FALSE)
training2 <- training[inTrain, ]
validation <- training[-inTrain, ]
```

Data Cleaning

This section removes certain columns that would interfere with the analysis. It looks for columns that have NAs and removes those columns, as well as removing the first two column. Additionally, all the columns are transformed into numeric, in an effort to make sure that all Data Sets are of the same type.

```
keep_cols<-colSums(is.na(training2))==FALSE
training3<-training2[,keep_cols]
training4<-training3[,-c(1,2,5,6)]
validation2<-validation[,keep_cols]
validation3<-validation2[,-c(1,2,5,6)]
testing2<-testing[,keep_cols]
testing3<-testing2[,-c(1,2,5,6)]

cols=c(1:55)
training4[,cols]=apply(training4[,cols],2,function(x) as.numeric(x))
validation3[,cols]=apply(validation3[,cols],2,function(x) as.numeric(x))
testing3[,cols]=apply(testing3[,cols],2,function(x) as.numeric(x))
testing3$problem_id<-as.factor(testing3$problem_id)
```

Fitting Models

Fitting a Random Forest model, tried different models and found the model use to be the one that both calculates fastest, and provides the best result.

```
rf_model<- randomForest(classe ~. , data=training4)
```

Making Prediction with Validation Data

Model tested on Validation Data

```
pred_val_rf <- predict(rf_model, validation3[,-58], type = "class")
```

Confusion Matrix

The confusion matrix below shows model accuracy. The resluts are very good, hence we believe the process undertaken is sufficient for the problem at hand.

```
cm_val_rf<-confusionMatrix(pred_val_rf, validation3$classe)
cm_val_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    2    0    0    0
##          B    0 1137    4    0    0
##          C    0    0 1022    2    0
##          D    0    0    0  962    0
##          E    0    0    0    0 1082
##
## Overall Statistics
##
```

```
##                Accuracy : 0.9986
##                  95% CI : (0.9973, 0.9994)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9983
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9982   0.9961   0.9979   1.0000
## Specificity            0.9995   0.9992   0.9996   1.0000   1.0000
## Pos Pred Value         0.9988   0.9965   0.9980   1.0000   1.0000
## Neg Pred Value         1.0000   0.9996   0.9992   0.9996   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1932   0.1737   0.1635   0.1839
## Detection Prevalence   0.2848   0.1939   0.1740   0.1635   0.1839
## Balanced Accuracy      0.9998   0.9987   0.9978   0.9990   1.0000
```

Making Prediction for test data

Now that we have selected a model based on the Training set and tested in on the Validation set, we can confidently use it to predict the results of the Test set.

```
##  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
```