Practical Machine Learning Project

Keyur Kariya

27/06/2018

Introduction

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. 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

Data Analysis and Creating Predictors

1- Preparations for Analysis

Loading libraries

```
library(caret)
library(randomForest)
library(knitr)
library(ggplot2)
```

Downloading and loading data

```
trainlink <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
validationlink <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training_main <- read.csv(url(trainlink))
validation_main<- read.csv(url(validationlink))
#featurePlot()
```

Checking data

```
head(training_main)
summary(training_main)
```

Some columns have lots of NA or blank values.

2- Cleaining the data

Cleaning columns with more than %60 NA values

```
training_cleaned<-training_main[ , colSums(is.na(training_main)) < .6] #We are creating new variables
```

Removing Near Zero Variances

```r
training_NZV <- nearZeroVar(training_cleaned, saveMetrics=TRUE)
training_cleaned<- training_cleaned[,training_NZV$nzv==FALSE]
```

Removing first column which is test numbers and checking

```r
training_cleaned$X<-NULL
```

Checking again Cleaned Data for anomalies

```r
summary(training_cleaned)
```

We also need to clean validation set with same values

```r
validation<-validation_main[ , colSums(is.na(training_main)) < .6]
validation<- validation[,training_NZV$nzv==FALSE]
validation$X<-NULL
```

3- Creating a test set from our training Data

We are dividing our training data into two parts; training (0.70) and test (0.30) to test it before validation

```r
set.seed(32343)
inTrain <- createDataPartition(y=training_cleaned$classe, p=0.70, list=FALSE)
training<-training_cleaned[inTrain,] #creating training data set

testing<-training_cleaned[-inTrain,]

dim(testing)
```

```
## [1] 5885   58
```

```r
dim(training)
```

```
## [1] 13737    58
```

```r
dim(validation)
```

```
## [1] 20 58
```

4- Creating Models

a- Classification Tree

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

```
## Loading required package: rpart

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

```r
prediction_class <- predict(modelFit_class,newdata=testing)
confusionMatrix(prediction_class,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1374  599  170  332   81
```

```
##          B    0    0    0    0    0
##          C  295  540  856  632  520
##          D    0    0    0    0    0
##          E    5    0    0    0  481
##
## Overall Statistics
##
##                Accuracy : 0.4607
##                  95% CI : (0.4479, 0.4735)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3059
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8208   0.0000   0.8343   0.0000  0.44455
## Specificity            0.7193   1.0000   0.5911   1.0000  0.99896
## Pos Pred Value         0.5376      NaN   0.3011      NaN  0.98971
## Neg Pred Value         0.9099   0.8065   0.9441   0.8362  0.88868
## Prevalence             0.2845   0.1935   0.1743   0.1638  0.18386
## Detection Rate         0.2335   0.0000   0.1455   0.0000  0.08173
## Detection Prevalence   0.4343   0.0000   0.4831   0.0000  0.08258
## Balanced Accuracy      0.7700   0.5000   0.7127   0.5000  0.72175
```

With "classification Tree" model, accuracy is 0.46 on testing set

b- Random Forest

```
modelFit_rf <- randomForest(classe ~.,data=training)
prediction_rf <- predict(modelFit_rf,newdata=testing)
confusionMatrix(prediction_rf,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    1    0    0    0
##          B    0 1138    1    0    0
##          C    0    0 1024    1    0
##          D    0    0    1  962    1
##          E    0    0    0    1 1081
##
## Overall Statistics

##
##                Accuracy : 0.999
##                  95% CI : (0.9978, 0.9996)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9987
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9991   0.9981   0.9979   0.9991
## Specificity            0.9990   0.9990   0.9990   0.9996   0.9990
## Pos Pred Value         0.9994   0.9991   0.9990   0.9979   0.9991
## Neg Pred Value         1.0000   0.9998   0.9996   0.9996   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1934   0.1740   0.1635   0.1837
## Detection Prevalence   0.2846   0.1935   0.1742   0.1638   0.1839
## Balanced Accuracy      0.9999   0.9995   0.9989   0.9988   0.9994
```

With Random Forest model, accuracy is 0.9988 on testing set which is better than Classification Tree model.

Summary

We will select "Random Forest method for our validation set which has better prection on testing set.

Validation set results:

```
fixFrame <- head(training,1) #take first row of training set

fixFrame <- fixFrame[, -length(colnames(fixFrame))] #remove last column (classe)

validation1<-validation[,-58] #remove id from validation data set since it is not needed for

validation1 <- rbind(fixFrame, validation1) #add first row of training set to validation set,

validation1 <- validation1[-1,] #remove first row we added previously

validation_predicts<-predict(modelFit_rf,newdata=validation1) # run RF method and it works well

validation_predicts # print classe from prediction
```

```
##  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  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
```