project_mlearning

marina

February 19, 2016

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 preprocessing

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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har.

First, we download and read csv files:

```
#reading csv files
#options(install.packages.check.source = "no", "repos"="http://cran.us.r-project.org")
#install.packages(pkgs = "caret",dependencies = c("Depends", "Imports"))
training <- read.csv("pml-training.csv", row.names = 1)
testing <- read.csv("pml-testing.csv",  row.names = 1)
#str(testing)
#str(training)
```

Here is the list of all packages used for this analysis:

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(ggplot2)
library(lattice)
library(kernlab)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Next, we are going to clean up the data. First, all the variables with almost no variation are identified and removed:

```
#identifying columns to delete
set.seed(123)
todelete <- nearZeroVar(training)
#deleting columns
training_fixed <- training[-todelete]
testing_fixed <- testing[-todelete]
```

All the variables with all missing values are removed, as well as irrelevant variable not useful for our analysis - timestamp:

```
# Deleting columns with all missing values
training_fixed<-training_fixed[,colSums(is.na(training_fixed)) == 0]
testing_fixed <-testing_fixed[,colSums(is.na(testing_fixed)) == 0]
#deleting irrelevant variable
training_fixed$cvtd_timestamp<- NULL
testing_fixed$cvtd_timestamp<- NULL
```

Modeling

First, we are going to subdivide training dataset into training and validation (for cross validation) subsets. Assigning 80% of observations to training subset

```
#partitioning training dataset into training and validation subsets
partition <- createDataPartition(training_fixed$classe,p=.8,list=FALSE)
trainingfinal <- training_fixed[partition,]
crossvalidation <- training_fixed[-partition,]
```

Next, we are going to fit the random forest model and calculate predicted values

```
#building random forest model
randomforestmodel <- randomForest(classe ~ ., data = trainingfinal)
predicted_values<-predict(randomforestmodel,trainingfinal)
```

Below is the confusion matrix for training dataset:

```
#printing confusion matrix
print(confusionMatrix(predicted_values, trainingfinal$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 4464    0    0    0    0
##          B    0 3038    0    0    0
##          C    0    0 2738    0    0
##          D    0    0    0 2573    0
##          E    0    0    0    0 2886
##
```

```
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Accuracy is very high for training dataset. Next stage is cross validation. We are going to predict values from validation dataset using our model and print out the confusion matrix:

```
#cross validating
predicted_validation<-predict(randomforestmodel,crossvalidation )
print(confusionMatrix(predicted_validation, crossvalidation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    0    0    0    0
##          B    0  759    0    0    0
##          C    0    0  684    1    0
##          D    0    0    0  642    1
##          E    0    0    0    0  720
##
## Overall Statistics
##
##                Accuracy : 0.9995
##                  95% CI : (0.9982, 0.9999)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9994
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   0.9984   0.9986
```

```
## Specificity            1.0000   1.0000   0.9997   0.9997   1.0000
## Pos Pred Value         1.0000   1.0000   0.9985   0.9984   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   0.9997   0.9997
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1935   0.1744   0.1637   0.1835
## Detection Prevalence   0.2845   0.1935   0.1746   0.1639   0.1835
## Balanced Accuracy      1.0000   1.0000   0.9998   0.9991   0.9993
```

Accuracy is also high for cross validation. Therefore, the model is acceptable.

Prediction

The last stage is to predict classe variable for testing dataset using our model:

```
#testing
predicted_testing<-predict(randomforestmodel,testing_fixed )
predicted_testing
```

```
##  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
```