Practical Machine Learning Write Up

Juuso Viljanen

Sunday, May 24, 2015

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).

Goal

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Data cleaning and preparations

1.Load package, set working directory and read data

```
library(caret)

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

setwd("~/Coursera/Practical Machine Learning/Viikko 3/Write Up")
pmlTreeni<-read.csv("pml-training.csv", header=T, na.strings=c("NA",
"#DIV/0!"))
pmlTesti<-read.csv("pml-testing.csv", header=T, na.string=c("NA", "#DIV/0!"))
```

2.Remove columns with NA:s

```
TreeniNA<-apply(pmlTreeni, 2, function(x) { sum(is.na(x)) })
Treenivalidi <- pmlTreeni[, which(TreeniNA == 0)]

TreeniNA<-apply(pmlTesti, 2, function(x) { sum(is.na(x)) })
Testivalidi <- pmlTesti[, which(TreeniNA == 0)]
```

3.Take just the sensor values data

```
Treenivalitut<-Treenivalidi[,c(8:60)]
Testivalitut<-Testivalidi[,c(8:60)]
```

4.Data partitioning The cleaned data set were subsetted to get the test set. Partitioning was performed to obtain a 60% training set and a 40% test set.

```
Treenissä<-createDataPartition(y=Treenivalitut$classe, p=0.60, list=F)
treeni<-Treenivalitut[Treenissä,]
testi<-Treenivalitut[-Treenissä,]
```

Results

Random forest trees were built for the training dataset using cross-validation.

1. Random Forest trees

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

## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=27
## - Fold1: mtry=27
## + Fold1: mtry=52
## - Fold1: mtry=52
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=27
## - Fold2: mtry=27
## + Fold2: mtry=52
## - Fold2: mtry=52
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=27
## - Fold3: mtry=27
## + Fold3: mtry=52
## - Fold3: mtry=52
## + Fold4: mtry= 2
## - Fold4: mtry= 2
## + Fold4: mtry=27
## - Fold4: mtry=27
```

```
## + Fold4: mtry=52
## - Fold4: mtry=52
## + Fold5: mtry= 2
## - Fold5: mtry= 2
## + Fold5: mtry=27
## - Fold5: mtry=27
## + Fold5: mtry=52
## - Fold5: mtry=52
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set

## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 9420, 9422, 9420, 9420, 9422
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.9887903  0.9858186  0.001228563  0.001554786
##   27    0.9887054  0.9857111  0.001523183  0.001927028
##   52    0.9845446  0.9804463  0.001961923  0.002482233
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

2. Confusion Matrix and Statistics

```
ennuste<-predict(modFit, newdata=testi)
confusionMatrix(ennuste, testi$classe)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    6    0    0    0
##          B    0 1509   12    0    0
##          C    0    3 1351   23    0
##          D    0    0    5 1263    4
##          E    0    0    0    0 1438
##
## Overall Statistics
##
##                Accuracy : 0.9932
```

```
##                  95% CI : (0.9912, 0.9949)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9915
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9941   0.9876   0.9821   0.9972
## Specificity            0.9989   0.9981   0.9960   0.9986   1.0000
## Pos Pred Value         0.9973   0.9921   0.9811   0.9929   1.0000
## Neg Pred Value         1.0000   0.9986   0.9974   0.9965   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1923   0.1722   0.1610   0.1833
## Detection Prevalence   0.2852   0.1939   0.1755   0.1621   0.1833
## Balanced Accuracy      0.9995   0.9961   0.9918   0.9904   0.9986
```

3. Prediction output

```
ennuste20<-predict(modFit, newdata=Testivalitut)
ennuste20
```

```
##  [1] 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
```

Conclusions

By using 52 predictors for five classes using cross-validation at a 5-fold an accuracy of 99.2% with a 95% CI [0.99-0.994] was achieved. Kappa value was 0.9902.

Once the predictions were gained, below script was used to obtain sigle text files for course web site to complete the submission assignment.

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")

write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(ennuste20)
```