Coursera Machine Learning Project

Di Zhou

Monday, November 17, 2014

Summary

The goal of this project is to predict the manner in which they did the exercise.The data for this project come from this source: (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

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)

Package Loading

Load the packages `caret` and `randomForest` into R.

```
## Loading required package: lattice
## Loading required package: ggplot2
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Data Cleaning

Notice there are a lot of blanks and missing values in the training set, we will fill the blank with NAs and remove columns with lots of NAs.

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##  [7] "num_window"           "roll_belt"            "pitch_belt"
## [10] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
## [13] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [16] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [19] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [22] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [25] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [28] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [31] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [34] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [37] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [40] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [43] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [46] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [49] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [52] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [55] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [58] "magnet_forearm_y"     "magnet_forearm_z"     "classe"
```

There are also some unrelated variables for building the model, such as X, user_names, new_window, etc, we are removing them.

Model Building

First, for cross validation, we are spliting the cleaned training set into 75% **Train** and 25% **Test** sets.

As recommended from the paper, we are using random forests method to build the model.

```
## Random Forest
##
## 14718 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 11774, 11774, 11775, 11774, 11775
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.9904199  0.9878808  0.001345832  0.001702528
##   27    0.9908955  0.9884831  0.002167313  0.002741061
##   52    0.9851880  0.9812619  0.002929520  0.003707450
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

After the model was built, we are using it to predict the **Test** set.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    2    0    0    0
##          B    6  939    4    0    0
##          C    0    3  850    2    0
##          D    0    0   13  790    1
##          E    0    0    4    0  897
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9901, 0.995)
##     No Information Rate : 0.2853
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9957   0.9947   0.9759   0.9975   0.9989
## Specificity            0.9994   0.9975   0.9988   0.9966   0.9990
## Pos Pred Value         0.9986   0.9895   0.9942   0.9826   0.9956
## Neg Pred Value         0.9983   0.9987   0.9948   0.9995   0.9998
## Prevalence             0.2853   0.1925   0.1776   0.1615   0.1831
## Detection Rate         0.2841   0.1915   0.1733   0.1611   0.1829
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9976   0.9961   0.9873   0.9970   0.9989
```

We expect the out of sample error is that: when we apply the model to Test set, the error between the predicted Test classe and the actual Test classe. We can see that the model for the Test set has 99.43% accuracy, so the out of sample error rate is 0.57%

Predict the testing set

The testing set was provided, and predicted using this model we get he answers listed below:

```
##  [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
```

The results are correct.