Practical Machine Learning Assignments

Kazuo Tamaru

22/2/2015

Our Goal

The goal of our project is to predict the manner in which participants did the exercise.

1.Load library & Get data

```
library(caret)
library(randomForest)
training_URL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test_URL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training <- read.csv(training_URL, na.strings = c("NA", "", "#DIV/0!"))
testing <- read.csv(test_URL, na.strings = c("NA", "", "#DIV/0!"))
```

2.Data Cleaning

Following variables are excluded from the analysis.
(1)Variables related to time and user information(column 1:6).
(2)Variables with at least one “NA”.

```
training <- training[, -c(1:6)]
testing <-testing[, -c(1:6)]
NoNA <- apply(training, 2, function(x) !any(is.na(x)))
training <- training[ ,NoNA]
testing <- testing[ ,NoNA]
dim(training)
```

```
## [1] 19622   54
```

3.Partitioning and Prediction

(1)We partition a 50% training set and a 50% test set.
(2)We use caret with randomForest as our model with 5 fold cross validation.

```
set.seed(24)
inTrain <- createDataPartition(y = training$classe, p = 0.5, list = FALSE)
Train <- training[inTrain, ]
Test <- training[-inTrain, ]
Model <- train(classe ~ ., data = Train, method="rf",
               trControl = trainControl(method = "cv", number = 5),
               prox = TRUE, allowParallel = TRUE)
Model
```

```
## Random Forest
##
## 9812 samples
##   53 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 7850, 7849, 7851, 7850, 7848
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa   Accuracy SD  Kappa SD
##    2    0.9899    0.9872  0.002735     0.003462
##   27    0.9953    0.9941  0.002085     0.002638
##   53    0.9934    0.9916  0.002195     0.002777
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

4.Evaluating the Model

ConfusionMatrix is as follows.
Accuracy is 99.6%. 95% Confidence Interval is (99.4%, 0.99.7%).

```
confusionMatrix(predict(Model, newdata = Test), Test$classe)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##         A 2790   12    0    0    0
##         B    0 1883    2    0    0
##         C    0    3 1709    7    0
##         D    0    0    0 1600   18
##         E    0    0    0    1 1785
##
## Overall Statistics
##
##               Accuracy : 0.996
##                 95% CI : (0.994, 0.997)
##    No Information Rate : 0.284
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.994
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.000    0.992    0.999    0.995    0.990
## Specificity            0.998    1.000    0.999    0.998    1.000
## Pos Pred Value         0.996    0.999    0.994    0.989    0.999
## Neg Pred Value         1.000    0.998    1.000    0.999    0.998
## Prevalence             0.284    0.193    0.174    0.164    0.184
## Detection Rate         0.284    0.192    0.174    0.163    0.182
## Detection Prevalence   0.286    0.192    0.175    0.165    0.182
## Balanced Accuracy      0.999    0.996    0.999    0.996    0.995
```

5.Predict classe

```
predict(Model, newdata = testing)
```

```
##  [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
```