Machine Learning : Predict Activity

Jenny Chen

May 22, 2016

1.Overview

In this project, we take the dataset from activity sensor device and build up a algoritm to predict the "classes" variable with given dataset.

2.Load and Clean 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

- source: http://groupware.les.inf.puc-rio.br/har.

```
# Load Library
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(rpart)
library(rpart.plot)

# Load and Read Dataset
train<-read.csv("~/R/data/ML/Train.csv",na.strings=c("#DIV/0","", "NA"))
test<-read.csv("~/R/data/ML/Test.csv",na.strings=c("#DIV/0","", "NA"))

# Clean data
## Removing Zero Covariates
nzv_train <- nearZeroVar(train, saveMetrics=TRUE)
train <- train[,nzv_train$nzv==FALSE & nzv_train$zeroVar==FALSE]
test <- test[,nzv_train$nzv==FALSE & nzv_train$zeroVar==FALSE]

## Remove any columns with more than 50% NAs.
Good <- lapply(train, function(x) sum(is.na(x)) / length(x)) <= 0.5
train <- train[ Good ]
test  <- test[ Good ]
```

3.Partition training dataset for cross validation

```
# Split Train into training and testing subset for cross validation
set.seed(888)
inTrain <- createDataPartition(y=train$classe, p=0.6, list=FALSE)
t_train <- train[inTrain, ]
t_test  <- train[-inTrain, ]
```

4.Build prediction model:decision tree

```
# Build model
Fit <- rpart(classe ~ ., data=t_train, method="class")

# Cross validation
x <- predict(Fit, t_test, type = "class")
y<-t_test$classe
table(x,y)
```

```
##    y
## x      A    B    C    D    E
##   A 2232    0    0    0    0
##   B    0 1518    0    0    0
##   C    0    0 1368    0    0
##   D    0    0    0 1286    0
##   E    0    0    0    0 1442
```

```
confusionMatrix(x,y)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    0 1518    0    0    0
##          C    0    0 1368    0    0
##          D    0    0    0 1286    0
##          E    0    0    0    0 1442
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9995, 1)
##     No Information Rate : 0.2845
##     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             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

5.Apply to Testing dataset

```
z<-predict(Fit,newdata = test)
```