C8project

Practical Machine Learning Course Project

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 (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (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 (http://groupware.les.inf.puc-rio.br/har). If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

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.

Initialization

```
library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
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
```

Get the data, Read in the training and testing dataset

```
set.seed(123)
trainUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

training <- read.csv(url(trainUrl), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(testUrl), na.strings=c("NA","#DIV/0!",""))
```

Data preparation and cleaning

Partion the training dataset into 2

```
inTrain <- createDataPartition(y=training$classe, p=0.7, list=FALSE)
myTraining <- training[inTrain, ]
myTesting <- training[-inTrain, ]
dim(myTraining)
```

```
## [1] 13737   160
```

```
dim(myTesting)
```

```
## [1] 5885  160
```

Remove the first few columns of the data

such as subject identifications, non movement activities related informations

```
myTraining <- myTraining[-c(1:5)]
myTesting <- myTesting[-c(1:5)]
```

Identify near zero variance predictors and remove the near zero variance predictors

```
near0 <- nearZeroVar(myTraining, saveMetrics=TRUE)
myTraining <- myTraining[,near0$nzv==FALSE]
```

Remove columns with more than 95% NA values

```
MostNA  <- sapply(myTraining, function(x) mean(is.na(x))) > 0.95
myTraining <- myTraining[, MostNA==FALSE]
clnames <- colnames(myTraining)
myTesting <- myTesting[,clnames]
```

Model building

```
set.seed(123)
```

Set up for Cross validation, using 10 fold validation

```
control <- trainControl(method='cv', number = 10)
```

Build model using Decision tree, Random Forest and Stochastic gradient boosting trees (gbm)
Decision Tree

```
myfit1 <- train(classe ~ ., data=myTraining,method='rpart',trControl=control)
```

Random forest

```
myfit2 <- train(classe ~ ., data=myTraining,method='rf',ntree=50,trControl=control)
```

Stochastic gradient boosting trees (gbm)

```
myfit3 <- train(classe ~ ., data=myTraining,method='gbm',trControl=control)
```

```
## Loading required package: gbm
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

```
## Loading required package: plyr
```

Model Assesment

```
pred1 <- predict(myfit1, newdata=myTesting)
cmatrix1 <- confusionMatrix(pred1, myTesting$classe)
pred2 <- predict(myfit2, newdata=myTesting)
cmatrix2 <- confusionMatrix(pred2, myTesting$classe)
pred3 <- predict(myfit3, newdata=myTesting)
cmatrix3 <- confusionMatrix(pred3, myTesting$classe)

Accuracy <- data.frame(
  Model = c('DT', 'RF', 'GBM'),
  Accuracy = rbind(cmatrix1$overall[1], cmatrix2$overall[1],cmatrix3$overall[1])
)
Accuracy
```

```
##   Model  Accuracy
## 1    DT 0.5984707
## 2    RF 0.9971113
## 3   GBM 0.9886151
```

The random forest model has the highest accuracy rate, as 0.9971
## Prediction with test dataset

```
model.predict <- predict(myfit2, testing)
```

Result

The confusion matrics shows Randone forest is the best model, It's prediction accuracy is 0.9971. The out of sample error estimation is 2.9%. The following are the most important variables are listed below.

```
varImp(myfit2)
```

```
## rf variable importance
## 
##   only 20 most important variables shown (out of 53)
## 
##                      Overall
## num_window           100.000
## roll_belt             61.841
## pitch_forearm         39.847
## yaw_belt              31.986
## magnet_dumbbell_z     27.804
## magnet_dumbbell_y     25.435
## pitch_belt            24.149
## roll_forearm          17.803
## roll_dumbbell         13.270
## magnet_dumbbell_x     10.930
## accel_dumbbell_y      10.639
## accel_forearm_x       10.446
## accel_belt_z           8.866
## total_accel_dumbbell   8.549
## magnet_belt_y          8.088
## accel_dumbbell_z       7.185
## magnet_belt_z          6.984
## magnet_forearm_z       6.480
## magnet_belt_x          5.633
## yaw_dumbbell           4.867
```