Machine learning course project

Juliet Nantege

8/27/2017

Overview

This document was compiled as part of the Machine learning Course project- a course offered by Johns Hopkins University on coursera. The project's goal is to predict the "classe" variable for the test data given the training data. The data is accessible and publicly available on this site http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

Reading and cleaning data

As per assignment requirements, data was downloaded from https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv and https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv for the training and test datasets respectively. I cleaned the data by replacing all empty fields with NAs, removing all columns with NA values and getting rid of descriptive unnecessary variables in columns 1 to 7. This left a total of 53 variables including the "classe" variable to be predicted in the training data and the "problem_id" variable in the test data.

```r
trainData <- read.csv("pml-training.csv")
testData <- read.csv("pml-testing.csv")
#cleaning training data
trainData[trainData=="" | trainData == "#DIV/0!"]<-NA
trainnoNAs <- trainData[,colSums(is.na(trainData)) == 0]
trainClean <- trainnoNAs[,-c(1:7)]

#cleaning test data
testData[testData=="" | testData == "#DIV/0!"]<-NA
testnoNAs <- testData[,colSums(is.na(testData)) == 0]
testClean <- testnoNAs[,-c(1:7)]
```

Partitioning and Model fitting

Using cross validation, I partition the training data into 2 subsets, one I will use for building the models and the other for testing and validating the models' accuracy before applying it to the test dataset. I fit 2 models using classification and regression trees (rpart) and Random forest (rf) methods and test their accuracy as shown below.

```r
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```r
library(rpart)
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(knitr)
set.seed(300)
intrain <- createDataPartition(trainClean$classe,p = 0.60)[[1]]
trainSub <- trainClean[intrain,]
validationSub <- trainClean[-intrain,]

#Rpart model fit
model1 <- train(classe ~., data = trainSub ,method = "rpart")
trainpred1 <- predict(model1, data = trainSub)
Validpred1 <- predict(model1, newdata = validationSub)
md1SelfAccuracy <- confusionMatrix(trainpred1, trainSub$classe)$overall[1]
md1PredAccuracy <- confusionMatrix(Validpred1, validationSub$classe)$overall[1]

#Random Forest model fit
model2 <- randomForest(classe~., data = trainSub, importance=TRUE)
trainpred2 <- predict(model2, data = trainSub)
Validpred2 <- predict(model2, newdata = validationSub)
md2SelfAccuracy <- confusionMatrix(trainpred2, trainSub$classe)$overall[1]
md2PredAccuracy <- confusionMatrix(Validpred2, validationSub$classe)$overall[1]

#Accuracy comparison
TrainData <- c(md1SelfAccuracy, md2SelfAccuracy)
Validation <- c(md1PredAccuracy, md2PredAccuracy)
Accuracy <- rbind(TrainData,Validation)
colnames(Accuracy) <- c("Reg Trees (rpart)", "Random Forest")
kable(Accuracy, align = "l", caption = "Table1: Accuracy comparison for both models when applied to trai
```

Table 1: Table1: Accuracy comparison for both models when applied to training sub dataset vs the validation dataset

	Reg Trees (rpart)	Random Forest
TrainData	0.4990659	0.9936311
Validation	0.4920979	0.9960489

- The Random Forest model (rf) is far more accurate with 99% level of accuracy for both the training subset and the validation data set as compared to Classification and Regression tree method (rpart) at 49%. So I decide to use the random forest (rf model) for my prediction.

Predicting on the test dataset

I choose to use model2(rf) for my testing prediction and it prooves highly accurate

```
#prediction
 testPrediction <- predict(model2, newdata = testClean[,-53])
 testPredictionResults <- data.frame("problem_id" = testClean$problem_id, "Classe_Prediction" = testPre
kable(testPredictionResults, align = "l", caption = "Table 2: Predicted classe for test dataset")
```

Table 2: Table 2: Predicted classe for test dataset

problem_id	Classe_Prediction
1	B
2	A
3	B
4	A
5	A
6	E
7	D
8	B
9	A
10	A
11	B
12	C
13	B
14	A
15	E
16	E
17	A
18	B
19	B
20	B

Conclusion

Random forests proved to be a highly superior method giving me the best model used to predict on the test data. Results are shown above in the table 2.