Activity Quality

Mario G.

17 06 2017

Note

Unfortunately an HTML conversion was not possible due to a cryptical error. Same with pdf. Only Word format worked. The script worked well outside of markdown :-(

Object of Study

More and more the internet of things enables us to document activities we are engaged with in our daily lives. While a lot of data are produced in this manner it is mainly the *quantity* of activities that attracts attention so far. This study deviates from this as our focus of attention will be the *quality* of involved activities: How well do participants perform barbell lifts? Available categories are classes A, B, C, D, and E.

Read Training Data

The data from this project come from this source: (http://groupware.les.inf.puc-rio.br/har) We begin with loading the training data.

```
mydata <- read.csv(file = 'pml-training.csv', header = TRUE,
                   na.strings=c("", " ","NA")) # read training data
dim(mydata)

## [1] 19622   160
```

Missing Values

```
nareport <- df_status(mydata)
mydata2 <- mydata[nareport$variable[nareport$q_na < 19216]]
mydata2 <- mydata2[,c(2,6:60)]
dim(mydata2)
```

The data has dimension 19622 rows * 160 columns. However, if we remove all variables that have more than 19215 NAs or empty fields we keep but 60 columns. One of these is an id which is also excluded as possible predictor. We also exclude three time stamps that seem unrelevant to our research question. So we'll keep one criterion (classe) and 55 possible predictors.

Divide Training Data for Cross Validation

The strategy will be to build a model with 50% of the training data and evaluate this model with the remaining 30% of the data. If the predictions are good enough the model will be used to predict the 20 cases of the testing file.

```
TrainSplit <- createDataPartition(y=mydata2$classe, p=0.7, list=FALSE)
Train <- mydata2[TrainSplit,]
Test <- mydata2[-TrainSplit,]
```

Model Construction

We'll be using a random forest to find an acceptable model.

```
modFit <- randomForest(classe ~ ., data=Train, method="rf")
modFit
```

Call: randomForest(formula = classe ~ ., data = Train, method = "rf") Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 7

```
    OOB estimate of  error rate: 0.27%
```

Confusion matrix: A B C D E class.error A 3905 0 0 0 1 0.0002560164 B 2 2654 2 0 0 0.0015048909 C 0 8 2388 0 0 0.0033388982 D 0 0 19 2233 0 0.0084369449 E 0 0 0 5 2520 0.0019801980

Model Evaluation

```
modPred <- predict(modFit, Test)
result <- confusionMatrix(modPred, Test$classe)
print(result)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    5    0    0    0
##          B    0 1134    2    0    0
##          C    0    0 1024    4    0
##          D    0    0    0  959    2
##          E    0    0    0    1 1080
## 
## Overall Statistics
## 
##                Accuracy : 0.9976
##                  95% CI : (0.996, 0.9987)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.997
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9956   0.9981   0.9948   0.9982
## Specificity            0.9988   0.9996   0.9992   0.9996   0.9998
## Pos Pred Value         0.9970   0.9982   0.9961   0.9979   0.9991
## Neg Pred Value         1.0000   0.9989   0.9996   0.9990   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1927   0.1740   0.1630   0.1835
## Detection Prevalence   0.2853   0.1930   0.1747   0.1633   0.1837
## Balanced Accuracy      0.9994   0.9976   0.9986   0.9972   0.9990
```

According to these crossvalidated results the prediction model is working extremely well.

Read Test Data

Now it's the turn of the real test data.

```r
mytestdata <- read.csv(file = 'pml-testing.csv',
                       header = TRUE, na.strings=c(""," ","NA")) # read test
data
library(data.table)
mytestdata2 <- mytestdata[,colnames(mydata2)[-56]] # classe is not available
in test data
```

Prediction

We'll use our crossvalidated model to forecast group membership.

```r
# handle bug in randomForest
mytestdata2 <- rbind(mydata2[1, -56] , mytestdata2)
mytestdata2 <- mytestdata2[-1,]

testPred <- predict(modFit, mytestdata2)
Prediction <- as.data.table(testPred)
Prediction[,casenum := 1:20]
library(knitr)
kable(Prediction)
```

testPred	casenum
B	1
A	2
B	3
A	4
A	5

E	6
D	7
B	8
A	9
A	10
B	11
C	12
B	13
A	14
E	15
E	16
A	17
B	18
B	19
B	20

And that's it!