ML assignment

Yalun

6/20/2020

Getting and cleaning the data

```
setwd("~/Downloads")
training <- read.csv("pml-training.csv",na.strings=c("NA","#DIV/0!", ""))
testing <- read.csv("pml-testing.csv",na.strings=c("NA","#DIV/0!", ""))

training<-training[,colSums(is.na(training)) == 0]
testing <-testing[,colSums(is.na(testing)) == 0]

training<-training[,8:60]
testing <-testing[,8:60]
```

training and testing data were extracted from csv files, and missing values was recognized as NAs. Then the columns with missing values were removed, and the data was further cleaned so that unrelated information (such as name and time) was not included.

Subset training data

You can also embed plots, for example:

```
library(caret)
```

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

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

```
sub <- createDataPartition(y=training$classe, p=0.75, list=FALSE)
trainingsub <- training[sub, ]
testingsub <- training[-sub, ]
```

the training data set was further divided into trainingsub and testingsub, so that our models could later be evaluated.

Model 1: decision tree

```
library(rpart)
model1 <- rpart(classe ~ ., data=trainingsub, method="class")
prediction1 <- predict(model1, testingsub, type = "class")
confusionMatrix(prediction1, testingsub$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
```

```
##          A 1233  160   21   56   17
##          B   52  519   38   58   73
##          C   42   87  695  130  104
##          D   47   67   52  501   65
##          E   21  116   49   59  642
##
## Overall Statistics
##
##                Accuracy : 0.7321
##                  95% CI : (0.7194, 0.7444)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6605
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8839   0.5469   0.8129   0.6231   0.7125
## Specificity            0.9276   0.9441   0.9103   0.9437   0.9388
## Pos Pred Value         0.8292   0.7014   0.6569   0.6844   0.7238
## Neg Pred Value         0.9526   0.8967   0.9584   0.9274   0.9355
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2514   0.1058   0.1417   0.1022   0.1309
## Detection Prevalence   0.3032   0.1509   0.2157   0.1493   0.1809
## Balanced Accuracy      0.9057   0.7455   0.8616   0.7834   0.8257
```

Model 2: random forest

```
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
model2 <- randomForest(classe ~. , data=trainingsub, method="class")
prediction2 <- predict(model2, testingsub, type = "class")
confusionMatrix(prediction2, testingsub$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    6    0    0    0
##          B    1  941    4    0    0
##          C    0    2  851    8    0
##          D    0    0    0  795    3
```

```
##          E    0    0    0    1  898
##
## Overall Statistics
##
##                Accuracy : 0.9949
##                  95% CI : (0.9925, 0.9967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9936
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   0.9916   0.9953   0.9888   0.9967
## Specificity            0.9983   0.9987   0.9975   0.9993   0.9998
## Pos Pred Value         0.9957   0.9947   0.9884   0.9962   0.9989
## Neg Pred Value         0.9997   0.9980   0.9990   0.9978   0.9993
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1919   0.1735   0.1621   0.1831
## Detection Prevalence   0.2855   0.1929   0.1756   0.1627   0.1833
## Balanced Accuracy      0.9988   0.9952   0.9964   0.9940   0.9982
```

Model3:SVM

```
library(e1071)
model3 <- svm(classe ~. , data=trainingsub, method="class")
prediction3 <- predict(model3, testingsub, type = "class")
confusionMatrix(prediction3, testingsub$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1384   70    1    4    0
##          B    5  853   21    1    9
##          C    5   25  828   99   18
##          D    0    0    3  700   11
##          E    1    1    2    0  863
##
## Overall Statistics
##
##                Accuracy : 0.9437
##                  95% CI : (0.9369, 0.95)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9287
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9921   0.8988   0.9684   0.8706   0.9578
## Specificity            0.9786   0.9909   0.9637   0.9966   0.9990
## Pos Pred Value         0.9486   0.9595   0.8492   0.9804   0.9954
## Neg Pred Value         0.9968   0.9761   0.9931   0.9752   0.9906
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2822   0.1739   0.1688   0.1427   0.1760
## Detection Prevalence   0.2975   0.1813   0.1988   0.1456   0.1768
## Balanced Accuracy      0.9854   0.9449   0.9661   0.9336   0.9784
```

Comparison

So far it seems that the three models each has accuracy of 73.94% (rpart), 99.37% (randomForest) and 94.47% (SVM), therefore I picked randomForest to run with our testing dataset.

```
testresult <- predict(model2, testing, type = "class")
testresult
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```