Machine Learning - Course Project

Overview

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. The purpose of this project is to predict the manner in which participants performed the exercises

Data Processing

Read the data, remove unneeded columns

```
library(randomForest)
library(caret)
set.seed(100) #reproducibility

data <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", na.strings = c

testdata <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", na.strings =
```

```
dim(data)
```

```
## [1] 19622   160
```

The table has 19622 rows. Take 80% of the data to use as training data. The remaining 20% is used as cross validation.

```
trainindex <- createDataPartition(data$classe, p = 0.8, list = FALSE)
trainfull <- data[trainindex,]
valset <- data[-trainindex,]
```

Use only the columns that have data in the set to be tested, and remove blanks.

```
traindata2 <- trainfull[,colSums(is.na(testdata)) == 0]
novartrain <- traindata2[ -c(1:7)]
valdata2 <- valset[,colSums(is.na(testdata)) == 0]
novarval <- valdata2[ -c(1:7)]
testdata2 <- testdata[ , colSums(is.na(testdata)) == 0]
novartestdata <- testdata2[ -c(1:7)]
```

Model

We use the randomForest method to build a model using the training data and show the prediction results, which should match since we are using the same data in both cases

```
model <- randomForest(classe ~., data = novartrain)

prediction<-predict(model,novartrain)

print(table(prediction,novartrain$classe))
```

```
##
## prediction    A    B    C    D    E
##          A 4464    0    0    0    0
##          B    0 3038    0    0    0
##          C    0    0 2738    0    0
##          D    0    0    0 2573    0
##          E    0    0    0    0 2886
```

Now we test against the 20% of the data set we held back for cross validation

```
validation<-predict(model,novarval)

print(table(validation,novarval$classe))
```

```
##
## validation    A    B    C    D    E
##          A 1116    4    0    0    0
##          B    0  752    3    0    0
##          C    0    3  680    8    0
##          D    0    0    1  635    0
##          E    0    0    0    0  721
```

```
confusionMatrix(validation,novarval$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    4    0    0    0
##          B    0  752    3    0    0
##          C    0    3  680    8    0
##          D    0    0    1  635    0
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.9952
##                  95% CI : (0.9924, 0.9971)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9939
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9908   0.9942   0.9876   1.0000
## Specificity            0.9986   0.9991   0.9966   0.9997   1.0000
## Pos Pred Value         0.9964   0.9960   0.9841   0.9984   1.0000
## Neg Pred Value         1.0000   0.9978   0.9988   0.9976   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1917   0.1733   0.1619   0.1838
## Detection Prevalence   0.2855   0.1925   0.1761   0.1621   0.1838
## Balanced Accuracy      0.9993   0.9949   0.9954   0.9936   1.0000
```

The cross validation accuracy is 99.4% so the out of sample error is therefore 0.6%.

Testing

Now we use the test data set with our model to try and predict excercise outcomes.

```
modelvalidation<-predict(model,novartestdata)

print(modelvalidation)
```

```
##  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
```