Practical Machine Learning Project

bdata7

September 24, 2017

Overview

The purpose of this project is to use data from accelerometers placed on the belt, forearm, arm and dumbell of subjects to predict the manner in which the participant performed a particular exercise.

In this report, a Random Forest model will be created and applied to make predictions about the data.

Read the Data into R

First, lets load the data into R. (Only the training dataset will be analyzed here; predictions will be made on the test dataset using the Random Forest model in a subsequent prediction quiz.)

```
training <- read.csv(url('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'))
testing <- read.csv(url('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'))
```

Prepare the data for modeling

Next, working only with the training dataset, the relevant data will be downselected and columns with NAs will be removed.

```
subsetTraining <- training[,-(1:7)]
subsetTraining <- subsetTraining[sapply(subsetTraining, function(x) !any(is.na(x)))]
```

Then, predictors with near zero variance are removed.

```
library(caret)
```

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

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

```
isNZV <- nearZeroVar(subsetTraining)
subsetTraining <- subsetTraining[,-isNZV]
```

Partition Training Data into Training and Test Sets

The training data is then split into a training dataset (70%) and a test dataset (30%).

```
set.seed(1234)
inTrain <- createDataPartition(subsetTraining$classe,p=0.7,list=FALSE)
trainData <- subsetTraining[inTrain,]
testData <- subsetTraining[-inTrain,]
```

Fit a Random Forest Model to the Training Dataset

Using the randomForest library, a Random Forest model will be fit to the training dataset. To speed up computational processing speed, the parallel and doParallel packages are used to pull in additional cores from the computer creating the Random Forest model.

Aside from the removal of near zero variance predictors (above), no additional preprocessing of the data was performed prior to creating the model. (This is one of the advantages to working with a Random Forest model - little or no preprocessing is generally needed.)

```
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(foreach)
library(parallel)
library(doParallel)
```

```
## Loading required package: iterators
```

```
c1 <- makeCluster(detectCores()-1)
registerDoParallel(c1)
rfModel <- randomForest(classe~.,
                        data=trainData,
                        importance=TRUE,
                        ntree=500)
stopCluster(c1)
registerDoSEQ()
```

Confusion Matrix

Using the other 30% of the training data that was set aside as a 'test dataset', one can determine how accurate the model is via a confusion matrix.

```
rfPred <- predict(rfModel,testData)
confusionMatrix(rfPred,testData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    6    0    0    0
##          B    0 1132    6    0    0
##          C    0    1 1020    4    0
##          D    0    0    0  959    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
```

```
##                Accuracy : 0.9969
##                  95% CI : (0.9952, 0.9982)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9961
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9939   0.9942   0.9948   1.0000
## Specificity            0.9986   0.9987   0.9990   1.0000   0.9998
## Pos Pred Value         0.9964   0.9947   0.9951   1.0000   0.9991
## Neg Pred Value         1.0000   0.9985   0.9988   0.9990   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1924   0.1733   0.1630   0.1839
## Detection Prevalence   0.2855   0.1934   0.1742   0.1630   0.1840
## Balanced Accuracy      0.9993   0.9963   0.9966   0.9974   0.9999
```

Conclusion

The Random Forest model can be used to give yield very accurate results (Balanced Accuracy > 0.99 for all 5 classes). No cross-validation or other pre-processing measures were used or needed, except for the removal of near zero variance predictors. Since the training data was partitioned such that only 70% of the data was used to build the model, the confusion matrix (testing the model against the 30% of data that was not used) suggests the expected out-of-sample error is $< 1\%$.