Machine Learning Project

C. Newcombe

Sunday, April 26, 2015

1. Build the Model

- Load the required libraries.

```
library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

- Download the training and test sets to the working directory.

```
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", destfile="trainpr
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", destfile="testproj
```

- Load the training and test data sets.

```
trainproj <- read.csv("trainproj.csv")
testproj <- read.csv("testproj.csv")
```

- Remove extra variables from the training set. We don't need any of the summary statistics (e.g., columns like min, max, var, stddev, etc.). Create a new data frame containing only the important variables (direct measurements and outcome columns).

```
train2 <- cbind(trainproj[, grep("^accel", names(trainproj))],
  trainproj[, grep("^magnet", names(trainproj))],
  trainproj[, grep("^roll", names(trainproj))],
  trainproj[, grep("^pitch", names(trainproj))],
  trainproj[, grep("^yaw", names(trainproj))],
  trainproj$classe)
colnames(train2)[37] <- "classe"
```

- Split `train2` dataset into a training set (80%) and a test set (20%) for cross-validation.

```
inTrain <- createDataPartition(y=train2$classe, p=0.8, list=FALSE)
train3 <- train2[inTrain,]
cvtest <- train2[-inTrain,]
```

- Create the random forest model.

```
model <- randomForest(classe ~ . , data=train3)
```

2. Cross Validation

- Compare the model prediction to the results of the `cvtest` data set.

```
cvpred <- predict(model, cvtest)
confusionMatrix(cvpred, cvtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    2    0    0    0
##          B    0  756    4    0    0
##          C    0    1  678    2    0
##          D    0    0    2  641    0
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.9972
##                  95% CI : (0.995, 0.9986)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9965
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9960   0.9912   0.9969   1.0000
## Specificity            0.9993   0.9987   0.9991   0.9994   1.0000
## Pos Pred Value         0.9982   0.9947   0.9956   0.9969   1.0000
## Neg Pred Value         1.0000   0.9991   0.9981   0.9994   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1927   0.1728   0.1634   0.1838
## Detection Prevalence   0.2850   0.1937   0.1736   0.1639   0.1838
## Balanced Accuracy      0.9996   0.9974   0.9952   0.9981   1.0000
```

3. Out-of-Sample Error

- Given by the `confusionMatrix()` calculation in #2, the out-of-sample accuracy is 99.52%. Therefore, the error is 0.48%.
- Given the accuracy/minimal error of the results from the cross-validation, the model type/variables do not need to be adjusted any further.

4. Predict Outcomes

- Use the `predict()` function with the test set.

```
predict(model, testproj)
```

```
##  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
```