Course Project

Daniel Traverso

6/14/2017

Loading the Data

Download pml-testing.csv and pml-training.csv into your current working directory

```
trainingData <- read.csv("pml-training.csv",na.strings=c("NA","#DIV/0!",""), header=TRUE)
testingData <- read.csv("pml-testing.csv",na.strings=c("NA","#DIV/0!",""), header=TRUE)
```

Clean the Data

Remove columns with all NA's

```
trainingData <- trainingData[, colSums(is.na(trainingData)) == 0]
testingData <- testingData[, colSums(is.na(testingData)) == 0]
```

We remove near-zero variance predictors from the dataset

```
library(caret)
```

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

```
badvar <- nearZeroVar(trainingData, saveMetrics=TRUE)
train <- trainingData[,badvar$nzv==FALSE]
test <- testingData[,badvar$nzv==FALSE]
```

Remove the row numbers, names of participants, and time stamps(first seven columns)

```
train <- train[-c(1:7)]
test <- test[-c(1:7)]
```

Data Partition

In order to get out-of-sample error we'll split the training set 70/30

```
set.seed(4321)
inTrain <- createDataPartition(train$classe, p = 0.7, list = FALSE)
train1 <- train[inTrain, ]
train2 <- train[-inTrain, ]
```

Build the Model

Next we build a decision tree on the training data using the caret package

```
control <- trainControl(method = "cv", number = 5)
model <- train(classe ~ ., data = train1, method = "rpart", trControl = control)
```

```
## Loading required package: rpart
```

```r
print(model, digits = 4)
```

```
## CART
##
## 13737 samples
##    51 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10990, 10989, 10990, 10991, 10988
## Resampling results across tuning parameters:
##
##   cp       Accuracy  Kappa
##   0.02767  0.4984    0.3442
##   0.03621  0.4795    0.3186
##   0.06365  0.3900    0.1754
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.02767.
```

Next we'll predict on the validation set

```r
pred <- predict(model, train2)
(confMatrix <- confusionMatrix(train2$classe, pred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1523   24   83   33   11
##          B  468  382  146  142    1
##          C  468   41  443   74    0
##          D  427  184  133  220    0
##          E  254  200  159  166  303
##
## Overall Statistics
##
##                Accuracy : 0.4879
##                  95% CI : (0.475, 0.5007)
##     No Information Rate : 0.5336
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3304
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4850  0.45969  0.45954  0.34646  0.96190
## Specificity            0.9450  0.85022  0.88153  0.85829  0.86014
## Pos Pred Value         0.9098  0.33538  0.43177  0.22822  0.28004
## Neg Pred Value         0.6160  0.90539  0.89278  0.91567  0.99750
## Prevalence             0.5336  0.14121  0.16381  0.10790  0.05353
## Detection Rate         0.2588  0.06491  0.07528  0.03738  0.05149
```

```
## Detection Prevalence   0.2845  0.19354  0.17434  0.16381  0.18386
## Balanced Accuracy      0.7150  0.65495  0.67054  0.60237  0.91102
```

```r
(accuracy <- confMatrix$overall[1])
```

```
##  Accuracy
## 0.4878505
```

The accuracy rate is 0.4879, so the out-of-sample error is 0.5121.

Test Set Predictions

Now we'll predict on the final holdout set

```r
(predict(model, test))
```

```
##  [1] C A D A A C D A A A C D C A D A E A A D
## Levels: A B C D E
```