Practical Machine Learning - Course Project

Kie Gouveia

July 26, 2015

Environment Set-Up

Download Data

```
fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
fileUrl2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"


if(!file.exists("./pml-training.csv")){
  download.file(fileUrl, "./pml-training.csv")
}

if(!file.exists("./pml-testing.csv")){
  download.file(fileUrl2, "./pml-testing.csv")
}
```

Set Seed for Reproducibility

```
set.seed(0012)
```

Load the training and test set

```
train0 <- read.csv("pml-training.csv", header = TRUE)
test0 <- read.csv("pml-testing.csv", header = TRUE)
```

Load necessary packages

```
library(caret)
```

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

```
library(rpart.plot)
```

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

Data Preparation

Data Cleaning and Variable Reduction

A few steps are taken to eliminate variables which contain significant amounts of missing data and to reduce the number of variables to only those which provide adequate new information. This helps with computational efficiency and overfitting.

1. Eliminate all variables where more than 50% of the data is NA.
2. Subset only numerical variables.
3. Reattach the classe variable to be predicted.

```
classe <- train0$classe
train <- train0[, colSums(is.na(train0)) < nrow(train0) * 0.5] # eliminate all variables which are more
train <- train[ , lapply(train, is.numeric) == TRUE]
train <- cbind(train, classe)
```

Note: I opted to remove non numeric variables such as the name of the participant. Although this may have improved the accuracy of teh model for the purposes of this project, it would not translate well to real-world application where the algorithm would be used on new participants.

Experimental Design

Sampling is used to assign 60% of dataset to a training set and 40% to a test set.

```
inTrain <- createDataPartition(y = train$classe,
                               p = 0.6,
                               list = FALSE)

trainSamp <- train[inTrain, ]
testSamp  <- train[-inTrain, ]
```

Analysis

A number of models were tested during the course of the assignment, including simple decision trees and gradient boosted models. The model which produced the best results during cross-validation was a Random Forest model which yielded an accuracy of NUMBER.

Random Forest

The model was preprocessed using Principle Components Analysis to remove variables which added little new information to the model (this also helped to ease computation time).

The model was trained on only the training data set.

```
fitControl <- trainControl(## 10-fold CV
                           method = "repeatedcv",
                           number = 10,
                           ## repeated ten times
                           repeats = 10)

rfFit1 <- train(classe ~ .,
                data = trainSamp,
                trControl = fitControl,
                preProcess = c("pca"),
                method = "rf")

print(rfFit1)
```

```
rfFit1 = readRDS("rfFit1.rds")
```

Cross Validation

To cross validate the data, we can run the final model on the testSamp dataset which we omitted while training the data.

```
rfPred <- predict(rfFit1, newdata = testSamp)
```

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

```
confusionMatrix(data = rfPred, testSamp$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2229    7    2    0    0
##          B    2 1505   15    0    0
##          C    0    6 1348    9    0
##          D    0    0    3 1277    4
##          E    1    0    0    0 1438
## 
## Overall Statistics
## 
##                Accuracy : 0.9938
##                  95% CI : (0.9918, 0.9954)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9921
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9987   0.9914   0.9854   0.9930   0.9972
## Specificity            0.9984   0.9973   0.9977   0.9989   0.9998
## Pos Pred Value         0.9960   0.9888   0.9890   0.9945   0.9993
## Neg Pred Value         0.9995   0.9979   0.9969   0.9986   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2841   0.1918   0.1718   0.1628   0.1833
## Detection Prevalence   0.2852   0.1940   0.1737   0.1637   0.1834
## Balanced Accuracy      0.9985   0.9944   0.9915   0.9960   0.9985
```

Based upon this cross validation, we expect the out of sample error to be 99.38%:

Prediction

Finally, I run the fest model on the test dataset which was provided for the assignment and assign it to variable answers.

Running best model, Random Forest, on test data

```
answers <- predict(rfFit1, newdata = test0)
```

Submission

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(answers)
```

Save Model for later use:

```
saveRDS(rfFit1, file = "rfFit1.rds")
```