Machine Learning Project

Mary Ann D

November 21, 2015

Executive Summary

The goal of the project is to predict the manner in which the participants in a fitness study performed their excercises. The data is provided in 2 files: training set and test set. The "classe" variable in the training set is the predictor. The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har.

Data Processing

Create the environment and load the training data set.

```r
library(knitr)
opts_chunk$set(fig.width=12,fig.height=6,cache=TRUE,cache.path="cache/",fig.path="Figs/")
library(caret)
```

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

```r
library(AppliedPredictiveModeling)
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```r
### Read the training data
trnfileUrl<- "pml-training.csv"
traindat<- read.csv(trnfileUrl, header=TRUE,sep = ",",na.strings = c("NA", "#DIV/0!","","-"))
```

Exploratory Data Analysis

After loading the training data set, review the data and transform and coerce the data as neccesary. The Data is a large data set of a mix of values and will be subsetted into a data set that is usable for fitting a model.

```r
removedat = is.na(traindat)      # Determine which variables are NA
keepcol<- nrow(traindat)*0.75
removecol<- which( colSums(removedat)> keepcol)  #Flag the columns that are more than 75% not applicabl
alldat = traindat[, -removecol]  # Remove those columns
alldat <- alldat[, -(1:7)]    # Remove the first 7 columns
table(sapply(alldat[1,], class))  # coerce the data
```

```
##
##  factor integer numeric
##       1      25      27
```

Build the Regression Models

Split the data

Split the training data and use the smaller portion as the Cross Validation data.

```
set.seed(1000)
inTrain <- createDataPartition(y=alldat$classe, p=0.6, list=FALSE)
mytrain <- alldat[inTrain, ]
myCV <- alldat[-inTrain, ]
```

Model Prediction

I used the random forest model since it the most widely used model for large data sets. Random forests correct for decision trees' habit of overfitting to their training set per the documentation.

```
modFit<- randomForest(classe ~., data= mytrain)
```

To see how accurate the model is, predict the model on the Cross Validation data.

```
predictrf<- predict(modFit, myCV, type="class")
confusionMatrix(predictrf, myCV$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2228   16    0    0    0
##          B    4 1497   10    0    0
##          C    0    5 1358   21    0
##          D    0    0    0 1265    3
##          E    0    0    0    0 1439
##
## Overall Statistics
##
##                Accuracy : 0.9925
##                  95% CI : (0.9903, 0.9943)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9905
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9862   0.9927   0.9837   0.9979
## Specificity            0.9971   0.9978   0.9960   0.9995   1.0000
## Pos Pred Value         0.9929   0.9907   0.9812   0.9976   1.0000
## Neg Pred Value         0.9993   0.9967   0.9985   0.9968   0.9995
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2840   0.1908   0.1731   0.1612   0.1834
## Detection Prevalence   0.2860   0.1926   0.1764   0.1616   0.1834
## Balanced Accuracy      0.9977   0.9920   0.9943   0.9916   0.9990
```

We see that this model is 99% accurate and the OOS(Out of Sample) rate is about .01%

Appendix

Apply the model to the Test data provided. Read and process the data in the same way that the training data was evaluated. Code provided

```r
TestfileUrl<- "pml-testing.csv"
testdat<- read.csv(TestfileUrl, header=TRUE,sep = ",",na.strings = c("NA", "#DIV/0!","","-"))
### Subset and transform the data
removedat = is.na(testdat)
keepcol<- nrow(testdat)*0.75
removecol<- which( colSums(removedat)> keepcol)
tstalldat = testdat[, -removecol]
tstalldat <- tstalldat[, -(1:7)]
table(sapply(tstalldat[1,], class))
```

```
##
## integer numeric
##      29      24
```

```r
### Run the model on the Test Data
predicttst<- predict(modFit, tstalldat, type="class")
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(predicttst)
```