PML Project

Practical Machine Learning Project (June 2014)

Introduction

This is the required report for the 'Practical Machine Learning' project. It describes a predictive model built using a collection of personal activity data, and the results of applying a test set of to the model to get predictions.

More details on the experiment that resulted in this data can be found here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

In the code below I have used the eval=FALSE option in some chunks so as to suppress evaluation, as it takes a long time and I have already run the analyses. I loaded up my prior analysis so as to print the results.

Data Load

Data used for this project were downloaded into a local directory from:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)
and
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Let's first load up the data from the local directory:

```
trn<-read.csv("pml-training.csv")
tst<-read.csv("pml-testing.csv")
```

Data Transformation

Inspection of the data using the summary function showed that the first seven variables in each dataset contained identification information not relevant to the analysis or prediction model. So we remove them. We also remove a number of variables with no values in them (all NAs), as they clearly have no predictive value. The remaining variables all report data from the activity monitors.

```
trn<-trn[,-c(1:7)]
tst<-tst[,-c(1:7)]
# Summary of the data shows several columns with only NAs in the training set
# Remove those columns from training
colsNA <- apply(trn,2,function(x) {sum(is.na(x))})
trn<-trn[,which(colsNA==0)]
# and from the test set
tst<-tst[,which(colsNA==0)]
```

Since we have such a large data set, let's take a random sample of 20% of the training set with the selection based on the distribution of the response variable (classe) in the training set.

```
# Train a simple tree with cross-validation
# Use a small sample (20%) to reduce compute time for CV tree
library(caTools)
set.seed(123)
split = sample.split(trn$classe, SplitRatio = 0.2)
tr1 = subset(trn, split==TRUE)
```

Build the model

We will build a simple tree model using randomForest, as this is the easiest and usually very accurate. We have cleaned up our data, and we can now just run the model with cross-validation with 3 folds. Note that we are predicting the 'classe' variable, using all the other variables in the cleaned up data.

```
# Train a simple tree with cross-validation
library(caret)
fC1<-trainControl(method="repeatedcv",number=3, repeats=3,verboseIter=TRUE, allowParallel=TRUE)
rf1<-train(classe~.,method="rf",data=tr1,trControl=fC1)
```

```
rf1
```

```
## Random Forest
##
## 3923 samples
##   85 predictors
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 3 times)
##
## Summary of sample sizes: 2615, 2615, 2616, 2616, 2615, 2615, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa  Accuracy SD  Kappa SD
##   2     0.3       0      1e-04        0
##   100   0.8       0.8    0.01         0.02
##   7000  1         0.9    0.008        0.01
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 6856.
```

The model report shows us the accuracy of the model is 95.9%, so we don't need to build any more models. We can apply the model to the training set, and take a look at the accuracy.

```
# check the accuracy on the training set
library(caret)
```

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

```
prf1<-predict(rf1,newdata=trn[,1:85])
```

```
## Loading required package: randomForest
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

```
confusionMatrix(trn$classe,prf1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5553   20    3    0    4
##          B   89 3641   55   11    1
##          C    1   49 3338   30    4
##          D    5   10   96 3090   15
##          E    1    6   12   13 3575
##
## Overall Statistics
##
##                Accuracy : 0.978
##                  95% CI : (0.976, 0.98)
##     No Information Rate : 0.288
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.973
##  Mcnemar's Test P-Value : 7.99e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.983    0.977    0.953    0.983    0.993
## Specificity             0.998    0.990    0.995    0.992    0.998
## Pos Pred Value          0.995    0.959    0.975    0.961    0.991
## Neg Pred Value          0.993    0.995    0.990    0.997    0.999
## Prevalence              0.288    0.190    0.179    0.160    0.183
## Detection Rate          0.283    0.186    0.170    0.157    0.182
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       0.991    0.984    0.974    0.988    0.996
```

And we can see from the confusion matrix statistics that the prediction accuracy is 97.83%. This suggests there is some over-fitting going on, however it is beyond the scope of this project to actually remove variables to build a realistic model.

We would expect the out-of-sample accuracy of the model to be below 97.83%, to account for overfitting in the model. Note that the model built by the experiment designers resulted in out-of-sample accuracy of between 70% and 80%.

Prediction on Test Set

We now apply the model to the test data.

```
# predict using the test set
prf1ts1<-predict(rf1,newdata=tst[,1:85])
```

Predicted data was applied to the course submission page, and the accuracy of the predictions was 100%. This is not surprising, as 1) the model is overfitted and 2) the size of test set is very small compared to the size of the training set.