Practical Machine Learning Assignment

Faisal Sardar

Sunday, March 22, 2015

Executive Summary

The assignment for the course was to take the data for personal activity devices, train a model and then provide projections on test set. To accomplish this the training and test sets were downloaded, data was cleaned and transformed for both sets simultaneously, the training set was then partitioned into a training and covalidation set with a 60/40 split. The training algorightm employed was random forests and numbers of trees was limited to 100 (to reduce the training time). The training model "modfit" was then applied to teh test set to perdict the "classe" for each of the 20 entries in the test set. Results submitted separately.

Cleaning Training & Testing data - Once the training and test files were downloaded the following transformations were applied to both the data sets. 1) Div/0, and blanks were converted to NA, 2) Columns consisting entirely of NA's were removed, 3) all the columns except for the first 8 were forced to type numeric, 4) Column 1,5 & 6 were removed (when applying the prediction model the types were conflicting with the trained model)

```r
pmltraining<-read.csv("./data/pml-training.csv", sep=",", header=TRUE,
                      na.strings=c("", "NA", "#DIV/0!"))
pmltesting<-read.csv("./data/pml-testing.csv", sep=",", header=TRUE,
                     na.strings=c("", "NA", "#DIV/0!"))
#remove NA columns from training set and also remove from test set
removeCol<-colSums(is.na(pmltraining))
pmltraining<-pmltraining[,removeCol<nrow(pmltraining)]
pmltesting<-pmltesting[,removeCol<nrow(pmltraining)]

#Set type of columns to numeric
for(i in c(8:ncol(pmltraining)-1)) {
    pmltraining[,i] = as.numeric(as.character(pmltraining[,i]))
}
for(i in c(8:ncol(pmltesting)-1)) {
    pmltesting[,i] = as.numeric(as.character(pmltesting[,i]))
}
#remove factor, character type columns - cause type issues with prediction model to test file
pmltraining<-pmltraining[,-c(1,5,6)]
pmltesting<-pmltesting[,-c(1,5,6)]

#identify and retain only columns with complete data to use in
isComplete<- function(x) {
    x[,sapply(x, function(y) !any(is.na(y)))]
}
incompl<- function(x) {
    names( x[,sapply(x, function(y) any(is.na(y)))] )
}

pmltraining <- isComplete(pmltraining)
pmltesting  <- isComplete(pmltesting)
```

Partitioning training set - Next step is to partition the training set to train and covalidate the results

```r
set.seed(55)

MLindex  <- createDataPartition(pmltraining$classe, p=.6, list=FALSE)
MLtrain <- pmltraining[MLindex,]
MLtest <- pmltraining[-MLindex,]
```

Train model - modfit - To train the model with y=classe, the method random forest was applied and the number of trees limited to 100 to limit processing time. The trained model “modfit” will be used for testing against the test partition data and the final test data.

```r
modfit <- train(MLtrain[,-57], MLtrain$classe, data=MLtrain, method="rf", ntree=100)
#save(modfit, file = "modfit.rda") #save model to avoid running every time
plot( varImp(modfit))
#load("modfit.rda")
```

Testing the modfit model on the partitioned data:

```r
confusionMatrix(predict(modfit,newdata=MLtest[,-57]),MLtest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    4    0    0    0
##          B    0 1514    2    0    0
##          C    0    0 1366    0    0
##          D    0    0    0 1285    0
##          E    0    0    0    1 1442
##
## Overall Statistics
##
##                Accuracy : 0.9991
##                  95% CI : (0.9982, 0.9996)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9989
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9985   0.9992   1.0000
## Specificity            0.9993   0.9997   1.0000   1.0000   0.9998
## Pos Pred Value         0.9982   0.9987   1.0000   1.0000   0.9993
## Neg Pred Value         1.0000   0.9994   0.9997   0.9998   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1930   0.1741   0.1638   0.1838
## Detection Prevalence   0.2850   0.1932   0.1741   0.1638   0.1839
## Balanced Accuracy      0.9996   0.9985   0.9993   0.9996   0.9999
```

Applying a confustionMatrix using the modfit model on the covalidation set gives us an accuracy level of .9991.

Code testing and generating the submission files using the modfit trained model against the assignment test data.

```
pmlpredict = function(x,i){
    predict(modfit, x[i,])
}

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