Courser Machine Learning Project

Introduction

The project predicts the manner in which test subjects performed exercise. The “classe” variable in the training set represents the outcome of the analysis. This project splits the train set into my.train and my.test data sets. The my.train data is used to build a predictive alhgorithm which predicts how the test subjects performed exercise, it is further tested on the my.test data before being applied to a separate parallel dataset called test.

The dataset contains data on six participants performing one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions:

- Class A exactly according to the specification
- Class B throwing the elbows to the front
- Class C lifting the dumbbell only halfway
- Class D lowering the dumbbell only halfway
- Class E throwing the hips to the front

This report describes how the model was built and cross validated. What the expected out of sample error is, and why the choices you did. The paper concludes by applying the prediction model to predict 20 different test cases.

This project http://groupware.les.inf.puc-rio.br/har#ixzz3ar6u7iCu (http://groupware.les.inf.puc-rio.br/har#ixzz3ar6u7iCu)

Load the data

```
train <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
test <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

Data Cleaning

The choice was made to remove variables with large numbers of NAs or missing values along with variables which do not relate to movement where removed from the dataset to enable a more effective machine learning algorithm.

```
set.seed(1234)
variables.with.nas = sapply(train, function(x) {sum(is.na(x))})
table(variables.with.nas)
```

```
## variables.with.nas
##     0 19216
##    93    67
```

```
variables.to.be.removed = names(variables.with.nas[variables.with.nas==19216])
train = train[, !names(train) %in% variables.to.be.removed]  #removes variables with NAs
train = train[,-c(1:7)]  # removes first seven variables, which do not relate to movement

troubling.variables.to.be.removed <- names(train) %in% c("kurtosis_roll_belt","kurtosis_picth_belt","kurtosis_yaw_belt","skewness_roll_belt","skewness_roll_belt.1","skewness_yaw_belt","max_yaw_belt","min_yaw_belt","amplitude_yaw_belt","skewness_yaw_arm", "skewness_pitch_arm","skewness_roll_arm","kurtosis_yaw_arm", "kurtosis_picth_arm", "kurtosis_roll_arm", "kurtosis_roll_dumbbell", "kurtosis_picth_dumbbell","kurtosis_yaw_dumbbell","skewness_roll_dumbbell", "skewness_pitch_dumbbell", "skewness_yaw_dumbbell", "max_yaw_dumbbell", "min_yaw_dumbbell", "amplitude_yaw_dumbbell", "kurtosis_roll_forearm", "kurtosis_picth_forearm","kurtosis_yaw_forearm","skewness_roll_forearm","skewness_pitch_forearm","skewness_yaw_forearm", "max_yaw_forearm", "min_yaw_forearm", "amplitude_yaw_forearm")

train <- train[!troubling.variables.to.be.removed]
```

Split data between train & test

The decision was made to partition the train data set into 90% my.train and 10% my.test. This enabled an effective predictive model to be created and tested before being applied to the test data set.

```
library(caret)
library(rpart)

inTrain <- createDataPartition(y=train$classe, p=0.9, list=FALSE)
my.train <- train[inTrain, ]
my.test <- train[-inTrain, ]
```

Build predictive algorithm

The predictive model was build using the rpart data function with method class. This allowed an effective predictive model to be created.

```
my.model = rpart(classe~., method="class", data=my.train)
```

Perform cross validation

The predictive model is used on the my.test data set and the model's accuracy assessed.

```
my.predictions <- predict(my.model, my.test, type="class")

confusionMatrix(my.predictions, my.test$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   A   B   C   D   E
##          A 493  53   3  23   7
##          B  26 246  33  21  24
##          C   8  35 276  52  40
##          D  13  29  23 206  17
##          E  18  16   7  19 272
## 
## Overall Statistics
##                                           
##                Accuracy : 0.7617          
##                  95% CI : (0.7422, 0.7804)
##     No Information Rate : 0.2847          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.6982          
##  Mcnemar's Test P-Value : 1.618e-08       
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8835   0.6491   0.8070   0.6417   0.7556
## Specificity            0.9387   0.9342   0.9166   0.9500   0.9625
## Pos Pred Value         0.8515   0.7029   0.6715   0.7153   0.8193
## Neg Pred Value         0.9529   0.9174   0.9574   0.9312   0.9459
## Prevalence             0.2847   0.1934   0.1745   0.1638   0.1837
## Detection Rate         0.2515   0.1255   0.1408   0.1051   0.1388
## Detection Prevalence   0.2954   0.1786   0.2097   0.1469   0.1694
## Balanced Accuracy      0.9111   0.7916   0.8618   0.7959   0.8590
```

```
saveRDS(my.model, "my.model.rds") # save my model to my working directory
```

The model has a predictive accuracy of 76%. Which means that it will predict one in four outcome incorrectly.

Rationalises variables in test data

Having created the predictive model on a subset of the data, the test data was subset with the same variables being retained as used in the predictiv emodel.

```
names.train<- names(train)  # selecting same variables in test data as train data
test <- test[,colnames(test) %in% names.train]
```

Write 20 prediction to text files

To complete the paper, the predictive model was used on the twenty samples in the test dataset and written to the working directory. The individual answer scripts where then uploaded into the Coursera website. The predictive model scored 15/20, that is it scored 75%. Which is frightenly close to the predictive model's forecats accuracy.

```
my.predictions.test <- predict(my.model, test, 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(my.predictions.test)
```

Conclusion

Making predictive models is fun, but I am a bit annoyed I lost 25% of the marks because my model wasn't up to getting 100% right!!!