Machine Learning Final Project

Introduction

The goal of the project is to predict the manner in which they did the exercise. You may use

Objective

The goal of the project is to predict the manner in which they did the exercise. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

About the Dataset

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

The model needs to predict the manner of exercise, given by the "classe" variable in the training set. There are **160 variables** in all and 19622 **data-points (rows)**.

The **test data with 20 cases** without the classe variabe serves as the final prediction performance test for the model.

Model Building

To build a prediction model, choices were made to ensure an elegant model, selecting variables most-likely to contribute individually to the model, selecting a model, cross validating it.

To build the model for predicting manner of exercise being performed, we rejected variables least related to exercise class.

The caret package was used for model training and improvement.

Choices made to approach the problem

Amongst the 159 variables, choices were made to eliminate variables likely unrelated to the exercise class variables.

- Variables with mostly "NA" data were eliminated with visual inspection of the training data as a table
- Individual components (x,y,z) are ignored in favor of **total_** type variables
- Variables measured in degrees were initially ignored
- Arm and forearm are common visual predictors of dumbell exercise manner per student's expert judgement

Initial predictor set: Total arm acceleration, total dumbell acceleration, total belt acceleration

Cross Validation

To validate the model before trying it on the supplied test data the supplied training data set was split into training and testing sets, in the ratio 7:3.

```
inTrain <- createDataPartition(y=training_supplied$classe, p=0.7, list=FALSE)
training <- training_supplied[inTrain,]
testing <- training_supplied[-inTrain,]
```

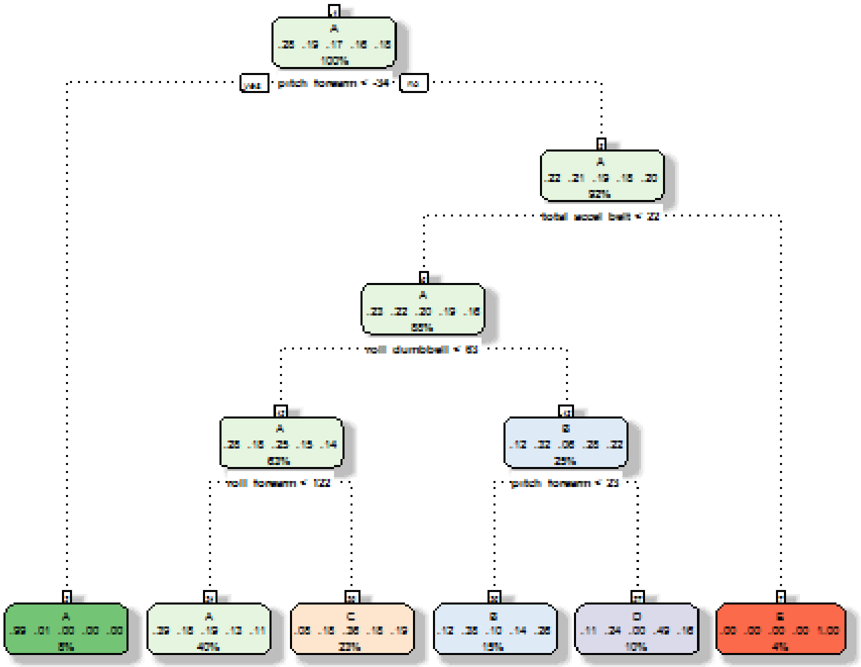
Preprocessing

As we trained the model preprocessing for scale and center to normalize the data did not increase the model accuracy significantly. So tree-based prediction was used **without preprocessing.**

Model training with Trees

```
modFit <- train(classe ~., data=training, method="rpart") # Trees (accuracy i
n 0.30s)
#modFit$finalModel

fancyRpartPlot(modFit$finalModel)
```



Rattle 2016-Mar-26 21:33:25 rdivecha

```
modFit$finalModel

## n= 13737 
## 
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
## 
##  1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)  
##    2) pitch_forearm< -33.95 1096    8 A (0.99 0.0073 0 0 0) *
##    3) pitch_forearm>=-33.95 12641 9823 A (0.22 0.21 0.19 0.18 0.2)  
##      6) total_accel_belt< 21.5 12056 9238 A (0.23 0.22 0.2 0.19 0.16)  
##       12) roll_dumbbell< 63.49108 8606 6188 A (0.28 0.18 0.25 0.15 0.14)  
##         24) roll_forearm< 122.5 5515 3354 A (0.39 0.18 0.19 0.13 0.11) *
##         25) roll_forearm>=122.5 3091 1963 C (0.083 0.18 0.36 0.18 0.19) *
##       13) roll_dumbbell>=63.49108 3450 2329 B (0.12 0.32 0.063 0.28 0.22)  
##         26) pitch_forearm< 22.75 2045 1265 B (0.12 0.38 0.1 0.14 0.26) *
##         27) pitch_forearm>=22.75 1405  721 D (0.11 0.24 0.0028 0.49 0.16) 
*
##      7) total_accel_belt>=21.5 585    0 E (0 0 0 0 1) *
```

Expected out of sample error rate

This is where we examine the fitness of the model to the data that was not used to train the model. The confusion matrix indicates the performance of the model internally.

Comparing predictions for the 30% data set aside with actual classifications, will reveal the efficacy of this model.

```
predicted_manner <- predict(modFit, newdata=testing)
#summary(predicted_manner)
confusionMatrix(testing$classe, predicted_manner)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1406   97  115   56    0
##          B  416  327  257  139    0
##          C  428  104  494    0    0
##          D  308  127  238  291    0
##          E  247  262  236   78  259
## 
## Overall Statistics
## 
##                Accuracy : 0.4719
##                  95% CI : (0.4591, 0.4847)
##     No Information Rate : 0.4766
##     P-Value [Acc > NIR] : 0.7715
## 
##                   Kappa : 0.3148
##  Mcnemar's Test P-Value : <2e-16
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.5012  0.35660  0.36866  0.51596  1.00000
## Specificity            0.9130  0.83655  0.88295  0.87352  0.85371
## Pos Pred Value         0.8399  0.28709  0.48148  0.30187  0.23937
## Neg Pred Value         0.6678  0.87568  0.82589  0.94452  1.00000
## Prevalence             0.4766  0.15582  0.22770  0.09584  0.04401
## Detection Rate         0.2389  0.05556  0.08394  0.04945  0.04401
## Detection Prevalence   0.2845  0.19354  0.17434  0.16381  0.18386
## Balanced Accuracy      0.7071  0.59658  0.62580  0.69474  0.92686
```

Prediction performance on the 20 test cases

```
predict(modFit, newdata=testing_supplied)
```

```
##  [1] C A C A A C C A A A B C C A C A A D A C
## Levels: A B C D E
```

Prediction using cross-validation

Using cross-validation training method K Nearest Neighbors **DOUBLES** model performance and scored **100% accurate on the quiz** associated with the project

```
modFitCV <- train(classe ~., data=training, method = "knn")
#modFitCV$finalModel
```

```
  predictCV <- predict(modFitCV, newdata=testing)
  confusionMatrix(predictCV,testing$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1614   50    8    1    3
##          B   33  987   27    8   41
##          C   10   65  944   51   34
##          D   12   12   29  867   55
##          E    5   25   18   37  949
## 
## Overall Statistics
##                                           
##                Accuracy : 0.911           
##                  95% CI : (0.9034, 0.9181)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.8874          
##  Mcnemar's Test P-Value : 5.287e-07       
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9642   0.8665   0.9201   0.8994   0.8771
## Specificity            0.9853   0.9770   0.9671   0.9781   0.9823
## Pos Pred Value         0.9630   0.9005   0.8551   0.8892   0.9178
## Neg Pred Value         0.9857   0.9683   0.9828   0.9802   0.9726
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2743   0.1677   0.1604   0.1473   0.1613
## Detection Prevalence   0.2848   0.1862   0.1876   0.1657   0.1757
## Balanced Accuracy      0.9747   0.9218   0.9436   0.9387   0.9297
```

Prediction performance on the 20 test cases

```
  predict(modFitCV, newdata=testing_supplied)
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Conclusion

Predicting if the manner of exercise was correct or incorrect required both instinct and experimentations through validation modeling. The tree-based modeling gave the opportunity to fine tune the variables, initially selected through instinct and common knowledge on what makes a dumbell exercise right or wrong.

The use of metrics in comparing relative performance of models is invaluable. K-folding technique revealed a far better prediction model compared to the popular random forest method.