Quality of Activity

Executive Summary

Using a Human Activity Benchmarking dataset created by 4 healthy subjects, this project builds a model to determine how a subject performs an excersize. There are 5 possible outcomes: sitting-down, standing-up, standing, walking, and sitting. A fitted model trained by data representing x, y, and z movements as recorded by various devices such as magnets and accelorators reveals the most optimal model is a Random Forest. The model here predicts how an excersize is performed with 99.45% accuracy. The model is then used to successfully predict results for 20 measures.

Pre-Processing

Data acquisition

Data for analysis is downloaded into the working directory and loaded into memory.

```
#test for existence to save download time from repeated runs of the project
if (!file.exists("training.csv"))
  download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv","training.csv")
if (!file.exists("testing.csv"))
  download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv","testing.csv")
#Read data into memory
training<-read.csv("training.csv", header=T,na.strings=c(""," ","na", "NA"))
testing<-read.csv("testing.csv", header=T,na.strings=c(""," ","na", "NA"))
```

Data processing

With data loaded, we build a tidy dataset by removing columns that are not valid for the model (too many NA values, not a measure of activity, etc).

```
#start a dataframe to hold the number of rows in the training & testing sets
tidyTrain<-data.frame(1:nrow(training))
tidyTest<-data.frame(1:nrow(testing))
#truncate the columns
tidyTrain<-tidyTrain[-1]
tidyTest<-tidyTest[-1]
#iterate through the training data
for (n in names(training)) {
  #determine the % of NA records
  x<-sum(is.na(training[n]))/nrow(training)
  #add the column to the tidy dataset if it has less than 90% NA values
  if (x<.9) {
    tidyTrain<-cbind(tidyTrain,training[n])
    #classe down not exist in the testing dataset, so avoid if the current column is classe
    if (n!="classe")
      tidyTest<-cbind(tidyTest,testing[n])
  }
}
```

```
#Columns 1:7 can be removed from the tidy dataset as they represent
#information about who and when the excersize was performed and are
#not a measure of how the excersize was performed
tidyTrain<-tidyTrain[,-c(1:7)]
tidyTest<-tidyTest[,-c(1:7)]
```

Develop training and test sets for cross validation

From the dataset available, create a training set using 75% of the data and a testing set using the remining 25%.

```
set.seed(12345)
#partition data 75/25 to build the model
inTrain<-createDataPartition(y=tidyTrain$classe,p=.75,list=F)
trainSet<-tidyTrain[inTrain,]
testSet<-tidyTrain[-inTrain,]
```

Modeling

Fitting various models

To derive the best model, compare results of Random Forest and Linear Discriminant Analysis. For each model, fit and predict

```
#Random Forest
fitRF<-randomForest(classe~.,data=trainSet,method="rf")
pRF<-predict(fitRF,newdata=testSet)
#LDA
fitLDA<-train(classe~.,data=trainSet,method="lda")
```

```
## Loading required package: MASS
```

```
pLDA<-predict(fitLDA,newdata=testSet)
```

Note: Generalized Linear Model is not compared as it is unable to determine final tuning parameters. In order to avoid overfitting, Naive-Bayes is not used.

Comparing the models

fitRF is expected to perform well, with an error rate of only .44%

```
fitRF
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = trainSet, method = "rf")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.4%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4184    1    0    0    0 0.0002389486
## B    8 2838    2    0    0 0.0035112360
## C    0   13 2553    1    0 0.0054538372
## D    0    0   21 2388    3 0.0099502488
## E    0    0    1    9 2696 0.0036954915
```

fitLDA is expected to only have 70.1% accuracy

```
fitLDA
```

```
## Linear Discriminant Analysis
##
## 14718 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 14718, 14718, 14718, 14718, 14718, 14718, ...
##
## Resampling results
##
##   Accuracy  Kappa      Accuracy SD  Kappa SD
##   0.70188   0.6227271  0.006453442  0.008052584
##
##
```

In comparing the tested models through a confusion matrix, Random Forest produces 99.45% accuracy (slightly under the .44% error rate) whereas LDA results in 69.8% accuracy (slightly under the expected 70.1% error rate).

```
#Random Forest
confusionMatrix(pRF,testSet$classe)$overall[1]
```

```
##  Accuracy
## 0.9944943
```

```r
#Linear Discriminant Analysis
confusionMatrix(pLDA,testSet$classe)$overall[1]
```

```
##  Accuracy
## 0.6980016
```

Choosing the best model

Due to its higher accuracy in testing, the Random Forest model is selected to predict the outcome of the input values in scope for this project. ##Predicting the outcome With the correct model identified, the following predicts the classe values using the Random Forest model.

```r
p<-predict(fitRF,newdata=tidyTest)
```

Prediction Results

The following are the prediction results

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

As a final step, results from running this model are also output to files in the working directory.

```
i<-0
for (v in p) {
  i<-i+1
  print(paste("saving to file: problem",i,".txt - answer: ",v))
  write(v,file=paste("problem",i,".txt"),append=F)
}
```

```
## [1] "saving to file: problem 1 .txt - answer:  B"
## [1] "saving to file: problem 2 .txt - answer:  A"
## [1] "saving to file: problem 3 .txt - answer:  B"
## [1] "saving to file: problem 4 .txt - answer:  A"
## [1] "saving to file: problem 5 .txt - answer:  A"
## [1] "saving to file: problem 6 .txt - answer:  E"
## [1] "saving to file: problem 7 .txt - answer:  D"
## [1] "saving to file: problem 8 .txt - answer:  B"
## [1] "saving to file: problem 9 .txt - answer:  A"
## [1] "saving to file: problem 10 .txt - answer:  A"
## [1] "saving to file: problem 11 .txt - answer:  B"
## [1] "saving to file: problem 12 .txt - answer:  C"
## [1] "saving to file: problem 13 .txt - answer:  B"
## [1] "saving to file: problem 14 .txt - answer:  A"
## [1] "saving to file: problem 15 .txt - answer:  E"
## [1] "saving to file: problem 16 .txt - answer:  E"
## [1] "saving to file: problem 17 .txt - answer:  A"
## [1] "saving to file: problem 18 .txt - answer:  B"
## [1] "saving to file: problem 19 .txt - answer:  B"
## [1] "saving to file: problem 20 .txt - answer:  B"
```

Citations

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. **Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements**. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6.

More information, including the datasets themselves, is available at the following location: http://groupware.les.inf.puc-rio.br/har