PML Prediction Report

Zhao JianZhuang

Summary

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. In this project, the main goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict how well they were doing the exercise using a relatively simple prediction model. The raw data can be obtained from http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises.

Processing Data

Read the data into R.

```
library (lattice)
library(ggplot2)
library(knitr)
library(caret)
library(corrplot)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
data_training <- read.csv("pml-training.csv", na.strings= c("NA","","  "))
data_testing <- read.csv("pml-testing.csv", na.strings= c("NA","","  "))
```

There are loads of NA values in the data. We need to clean and remove these columns from the data set. The first eight columns that acted as identifiers for the experiment were also removed.

```
training_NAs <- apply(data_training, 2, function(x) {sum(is.na(x))})
training_new <- data_training[,which(training_NAs == 0)]
testing_NAs <- apply(data_testing, 2, function(x) {sum(is.na(x))})
testing_new <- data_testing[,which(testing_NAs == 0)]
training_final <- training_new[8:length(training_new)]
testing_final  <- testing_new[8:length(testing_new)]
```

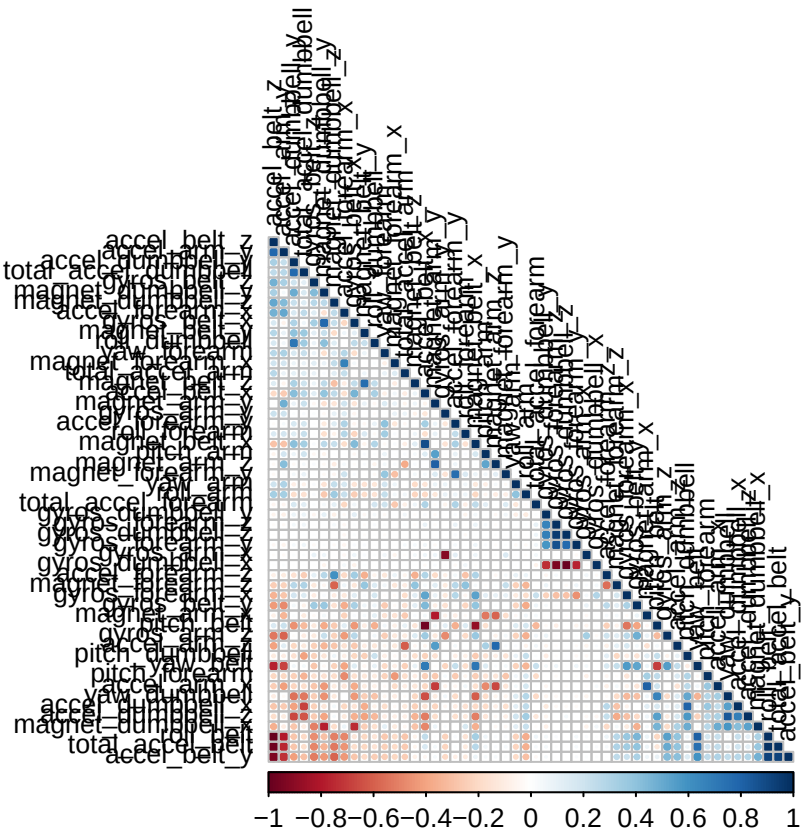
Training a Model

The test data set was split up into training and cross validation sets in a 70:30 ratio in order to train the model and then test it against data it was not specifically fitted to.

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

A random forest model was selected to predict the classification. The correllation plot was used to see how strong the variables relationships are with each other.

```
corMatrix <- cor(training[, -length(training)])
corrplot(corMatrix, order = "FPC", method = "circle", type = "lower", tl.cex = 0.8,  tl.col = rgb(0, 0,
```



In the graph, the dark red and blue colours indicate a highly negative and positive relationship respectively between the variables. There was not much concern for highly correlated predictors which mean that all of them can be contained in the model.

And then, a model was fitted with the outcome set to the training class and all the other variables used to predict.

```
model <- randomForest(classe ~ ., data = training)
model
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.5%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    2    0    0    0 0.0005120328
## B   11 2639    8    0    0 0.0071482318
## C    1   16 2379    0    0 0.0070951586
```

```
## D    0    0   21 2230    1 0.0097690941
## E    0    0    2    7 2516 0.0035643564
```

The model produced a very small OOB estimate of error rate of 0.5%. This was deemed good enough to progress the testing.

Model Cross Variation

The model was further used to classify the remaining 30% of cross validation data.

```
predictCross <- predict(model, crossval)
confusionMatrix(crossval$classe, predictCross)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    2    0    0    0
##          B    8 1128    3    0    0
##          C    0    5 1021    0    0
##          D    0    0    8  955    1
##          E    0    0    4    0 1078
##
## Overall Statistics
##
##                Accuracy : 0.9947
##                  95% CI : (0.9925, 0.9964)
##     No Information Rate : 0.2855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9933
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9952   0.9938   0.9855   1.0000   0.9991
## Specificity            0.9995   0.9977   0.9990   0.9982   0.9992
## Pos Pred Value         0.9988   0.9903   0.9951   0.9907   0.9963
## Neg Pred Value         0.9981   0.9985   0.9969   1.0000   0.9998
## Prevalence             0.2855   0.1929   0.1760   0.1623   0.1833
## Detection Rate         0.2841   0.1917   0.1735   0.1623   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9974   0.9958   0.9922   0.9991   0.9991
```

The confusion matrix and statistics shows that this model has a 99.51% prediction accuracy. Again, this model was proved good enough to predict new data.

Prediction of Testing Data

A separate data set was then loaded into R and cleaned in the same manner as before. The model was then used to predict the classifications of the 20 results of this new data.

```
predTesting <- predict(model, testing_final)
predTesting
```

```
##  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
```

Conclusion

With the abundance of information given from multiple measuring instruments it's possible to accurately predict how well a person is preforming an excercise using a relatively simple prediction model.