Prediction Assignment

Isis Ibarra

17/11/2018

1. Overview

People regularly quantify how much of an excercise they do, but rarely measure their performance. In this investigation, data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants will be used to predict the manner of the subject. Random forest, decision tree and generalized boosted model will be the methods implemented to determine the best predicion. R programming will be the major tool used in the project.

2. Data preparation

The data for this project is taken from the Human Activity Recognition project by Groupware@LES. For more information, please visit their website.

As the first step in this investigation, data preparation is needed. The following code is used to load the corresponding libraries.

```
library(knitr)
library(caret)
library(rpart)
library(rpart.plot)
library(rattle)
library(randomForest)
library(corrplot)
library(e1071)
```

The next step is loading the dataset from the URL provided, and store the information into the `training` and `testing` variables.

```
trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainFile <- "pml-traininig.csv"
testFile <- "pml-testing.csv"

if(!file.exists(trainFile))
{
  download.file(trainURL, destfile = trainFile)
}
if(!file.exists(testFile))
{
  download.file(testURL, destfile = testFile)
}

training <- read.csv(trainFile)
testing <- read.csv(testFile)
```

In order to have a better predictive model, `training` dataset is partitioned into 2 subsets:
* `trainSet`: consists of 70% of the dataset; will be used for the modeling process
* `validationSet`: consists of 30% of the dataset; will be used for cross validation

```
trainingPartition <- createDataPartition(training$classe, p = 0.7, list = FALSE)
trainSet <- training[trainingPartition, ]
validationSet <- training[-trainingPartition, ]
```

To ensure classification rules can be applied to the dataset, data cleansing must be done. The following considerations will be entered:
1. Remove the constant and almost constant variables accross the sample
2. Remove variables composed of at least 95% of missing values or empty strings
3. Remove identification variables, such as time and user information

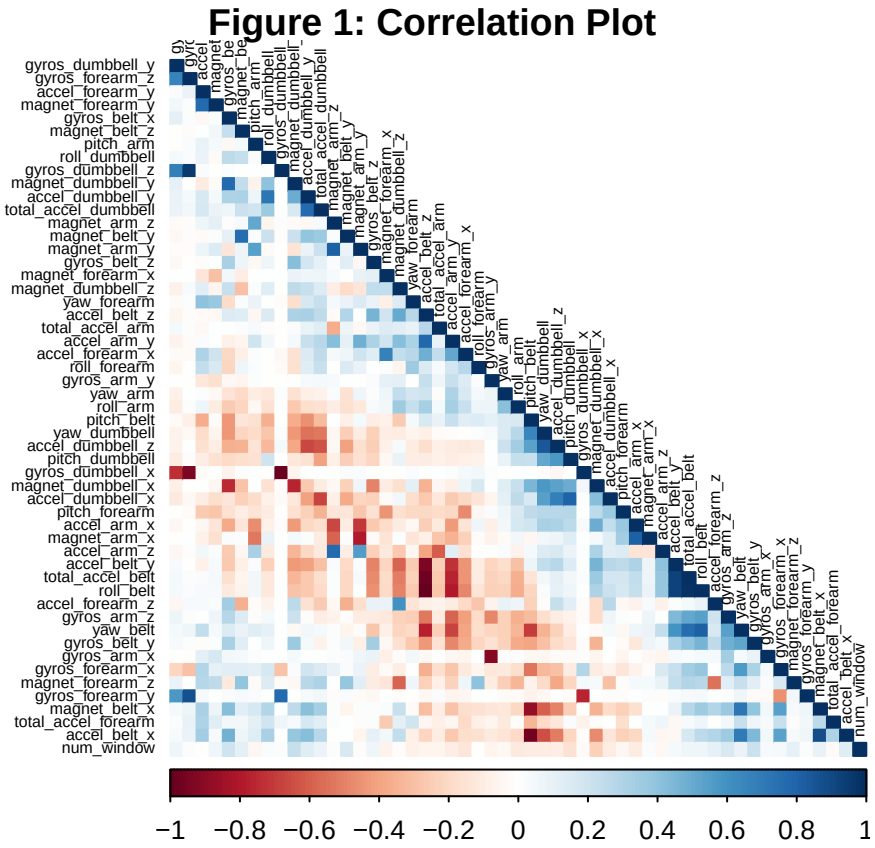
```
# Remove constant and almost constant varibales across the sample
NZV <- nearZeroVar(trainSet)
trainSet <- trainSet[, -NZV]
validationSet <- validationSet[, -NZV]
# Remove variables with mostly missing values
na <- sapply(trainSet, function(x) mean(is.na(x))) > 0.95
trainSet <- trainSet[, na == FALSE]
validationSet <- validationSet[, na == FALSE]
# Remove identification variables
trainSet <- trainSet[, -(1:5)]
validationSet  <- validationSet[, -(1:5)]
```

After this cleansing process, there are 53 variables suited for analysis.

3. Exploratory analysis

To get a better insight of the relationship between the variables, a correlation analyisis will be done.

```
plotCorrelation <- cor(trainSet[, -54])
corrplot(plotCorrelation, method = "color", order = "AOE", type = "lower", tl.cex = 0.5, tl.col = rgb(0
```



In Figure 1: Correlation Plot, highly possitively correlated values are painted in dark blue, while negatively are colored dark red.

4. Predictive models

Now, three popular methods will be applied to model the regressions in the training dataset. A confusion matrix is plotted at the end of each analysis to better visualize the accuracy of the models.

4.1 Random forest

```
# Set seed for reproducibility
set.seed(1234)
# Create random forest model
controlRF <- trainControl(method = "cv", number = 3, verboseIter = FALSE)
modelRF <- train(classe ~ ., data = trainSet, method = "rf", trControl = controlRF)
modelRF$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.27%
```

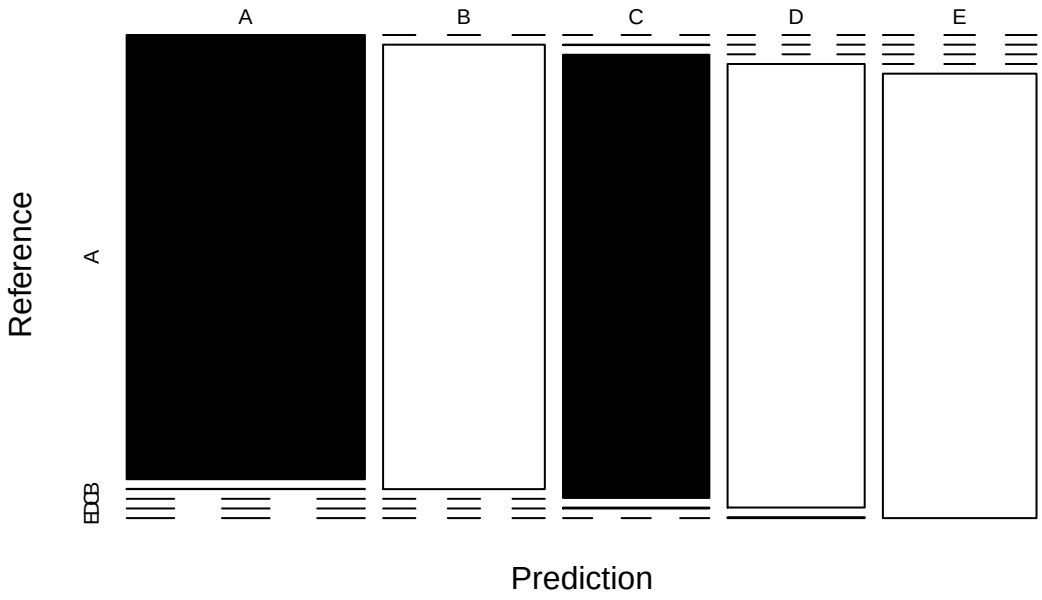
```
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3905    0    0    0    1 0.0002560164
## B    5 2649    3    1    0 0.0033860045
## C    0    6 2390    0    0 0.0025041736
## D    0    0   12 2239    1 0.0057726465
## E    0    1    0    7 2517 0.0031683168
```

```
# Predict using the test dataset
predictRF <- predict(modelRF, newdata = validationSet)
confusionMatrixRF <- confusionMatrix(predictRF, validationSet$classe)
confusionMatrixRF
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    1    0    0    0
##          B    0 1137    0    0    0
##          C    0    1 1026    2    0
##          D    0    0    0  962    2
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.999
##                  95% CI : (0.9978, 0.9996)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9987
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9982   1.0000   0.9979   0.9982
## Specificity            0.9998   1.0000   0.9994   0.9996   1.0000
## Pos Pred Value         0.9994   1.0000   0.9971   0.9979   1.0000
## Neg Pred Value         1.0000   0.9996   1.0000   0.9996   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1932   0.1743   0.1635   0.1835
## Detection Prevalence   0.2846   0.1932   0.1749   0.1638   0.1835
## Balanced Accuracy      0.9999   0.9991   0.9997   0.9988   0.9991
```

```
# Plot results
plot(confusionMatrixRF$table, col = confusionMatrixRF$byClass,
    main = paste("Figure 2: Random Forest Plot - Accuracy =",
                 round(confusionMatrixRF$overall['Accuracy'], 3)))
```

Figure 2: Random Forest Plot − Accuracy = 0.999

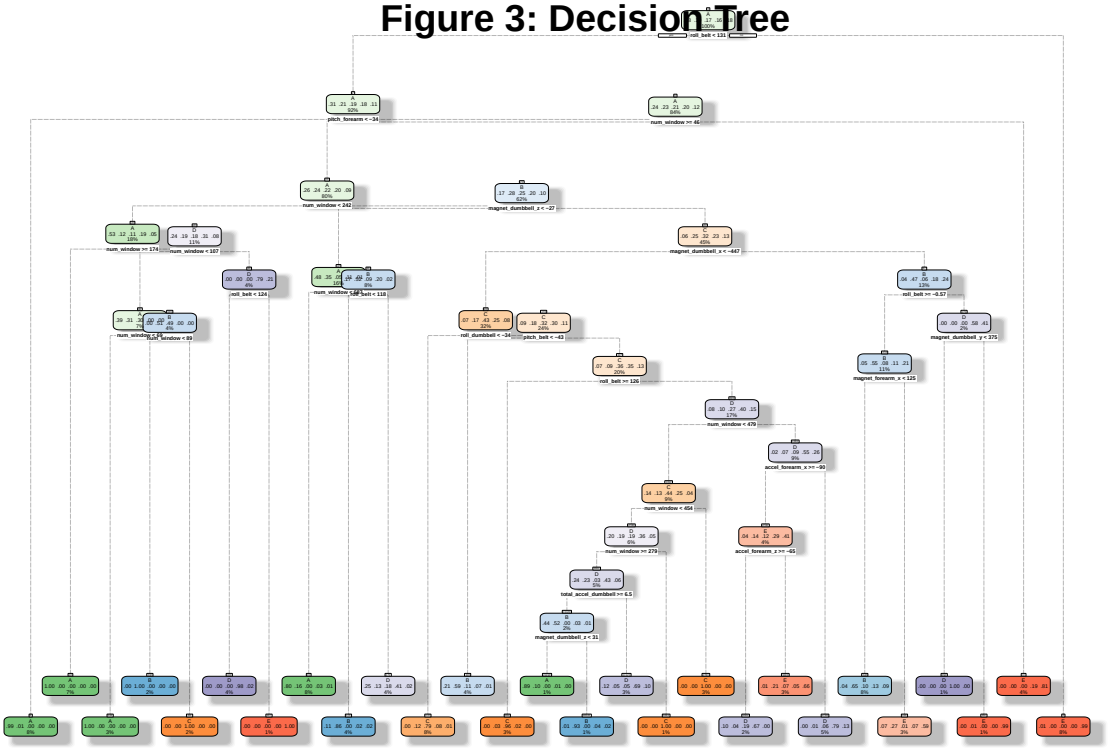


4.2 Decision tree

```
# Set seed for reproducibility
set.seed(1234)
# Create decision tree model
modelDT <- rpart(classe ~ ., data = trainSet, method = "class")
fancyRpartPlot(modelDT, main = "Figure 3: Decision Tree")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

Figure 3: Decision Tree



Rattle 2018−Nov−18 17:58:13 MariaGuadalupeVillarrealGarza

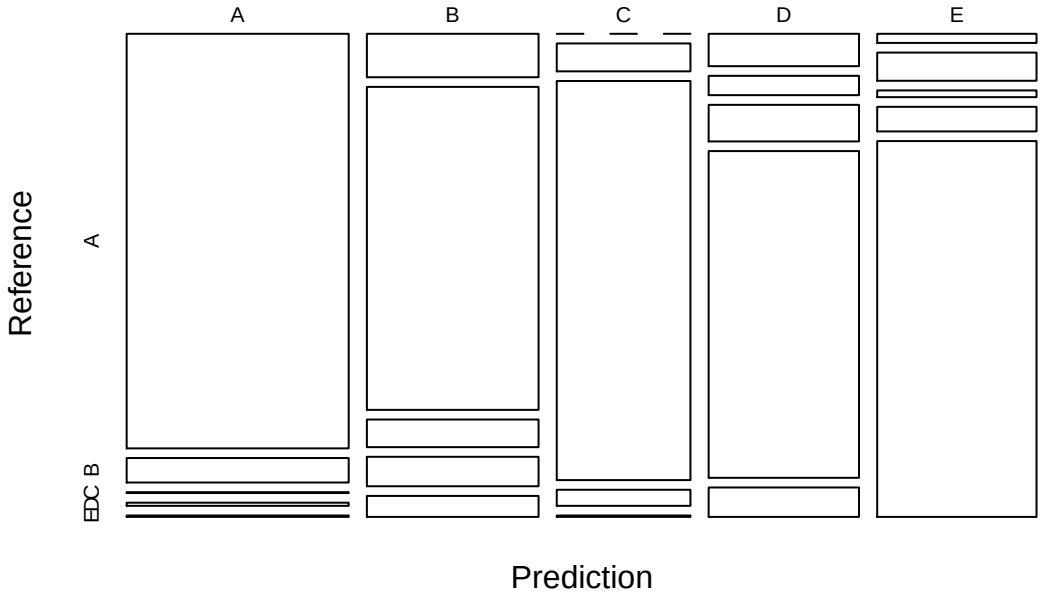
```
# Predict using the test dataset
predictDT <- predict(modelDT, newdata = validationSet, type = "class")
confusionMatrixDT <- confusionMatrix(predictDT, validationSet$classe)
confusionMatrixDT
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1456   86    3   11    5
##          B  118  877   75   80   57
##          C    0   59  844   34    3
##          D   77   46   87  777   70
##          E   23   71   17   62  947
##
## Overall Statistics
##
##                Accuracy : 0.8328
##                  95% CI : (0.823, 0.8422)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.789
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8698   0.7700   0.8226   0.8060   0.8752
```

```
## Specificity            0.9751   0.9305   0.9802   0.9431   0.9640
## Pos Pred Value         0.9327   0.7266   0.8979   0.7351   0.8455
## Neg Pred Value         0.9496   0.9440   0.9632   0.9613   0.9717
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2474   0.1490   0.1434   0.1320   0.1609
## Detection Prevalence   0.2653   0.2051   0.1597   0.1796   0.1903
## Balanced Accuracy      0.9224   0.8502   0.9014   0.8746   0.9196
```

```
# Plot results
plot(confusionMatrixDT$table, col = confusionMatrixDT$byClass,
    main = paste("Figure 4: Decision Tree Plot - Accuracy =",
                 round(confusionMatrixDT$overall['Accuracy'], 3)))
```

Figure 4: Decision Tree Plot − Accuracy = 0.833



4.3 Generalized boosted model

```
# Set seed for reproducibility
set.seed(1234)
# Create decision tree model
controlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
modelGBM <- train(classe ~ ., data = trainSet, method = "gbm", trControl = controlGBM, verbose = FALSE)
modelGBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 43 had non-zero influence.
```

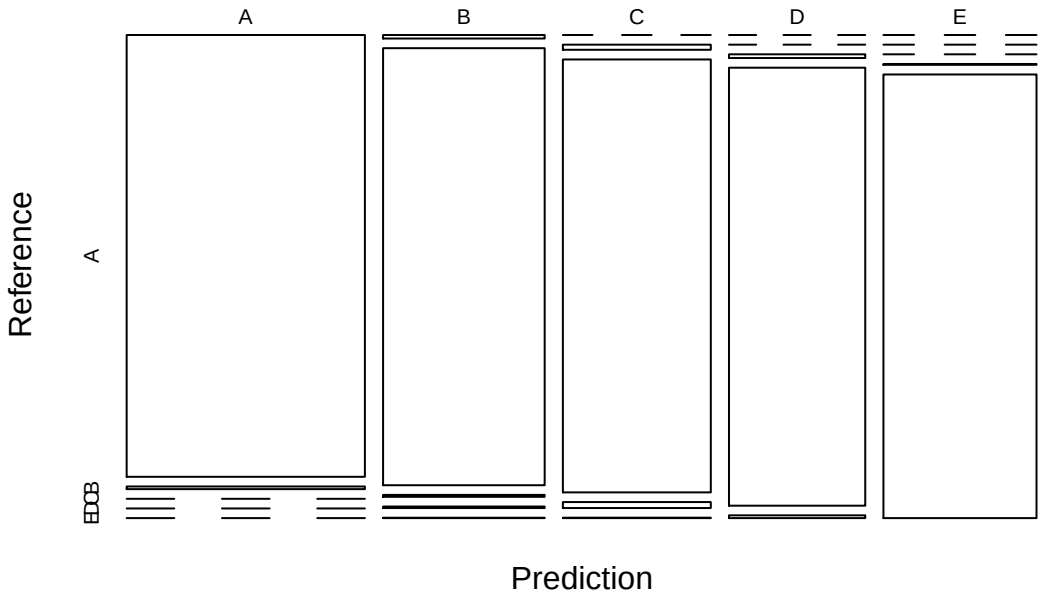
```
# Predict using the test dataset
predictGBM <- predict(modelGBM, newdata = validationSet)
confusionMatrixGBM <- confusionMatrix(predictGBM, validationSet$classe)
confusionMatrixGBM
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1665   10    0    0    0
##          B    9 1117    5    4    1
##          C    0   12 1013   14    1
##          D    0    0    8  944    6
##          E    0    0    0    2 1074
##
## Overall Statistics
##
##                Accuracy : 0.9878
##                  95% CI : (0.9846, 0.9904)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9845
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9946   0.9807   0.9873   0.9793   0.9926
## Specificity            0.9976   0.9960   0.9944   0.9972   0.9996
## Pos Pred Value         0.9940   0.9833   0.9740   0.9854   0.9981
## Neg Pred Value         0.9979   0.9954   0.9973   0.9959   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2829   0.1898   0.1721   0.1604   0.1825
## Detection Prevalence   0.2846   0.1930   0.1767   0.1628   0.1828
## Balanced Accuracy      0.9961   0.9883   0.9909   0.9882   0.9961
```

```r
# Plot results
plot(confusionMatrixGBM$table, col = confusionMatrixGBM$byClass,
    main = paste("Figure 5: Generalized Boosted Model Plot - Accuracy =",
                 round(confusionMatrixGBM$overall['Accuracy'], 3)))
```

Figure 5: Generalized Boosted Model Plot − Accuracy = 0.988



5. Applying selected model to test data

As for this investigation, the accuracy of the selected models is the following:
* Random forest: 0.999
* Decision tree: 0.729
* GBM: 0.989 Therefore, the random forest method must be used to prefict the results.

```
predict <- predict(modelRF, newdata = testing)
predict
```

```
##  [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
```