Practical Machine Learning Final Project

Arpad Kovacs

October 20, 2017

Human Activity Recognition

Prediction with Machine Learning Models

Background

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. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Setup R environment

Data Conditioning

This section describes the following steps.

1. Downloading the data
2. Cleaning the data

a. Removing variables that do not provide useful information
b. Removing variables with near zero variance
c. Removing variables that are mostly N/A
d. Removing variables that are highly correlated

3. Splitting the training dataset to training and validation

Downloading

The dataset were downloaded from the provided links.

```
#Web address of the files
TrainDL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
TestDL  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

#Download the datasets
train <- read.csv(url(TrainDL))
testing  <- read.csv(url(TestDL))
```

Cleaning

```
#Removing variables that are only for identification purposes (columns 1 through 5)
train <- train[, -(1:5)]
dim(train)
```

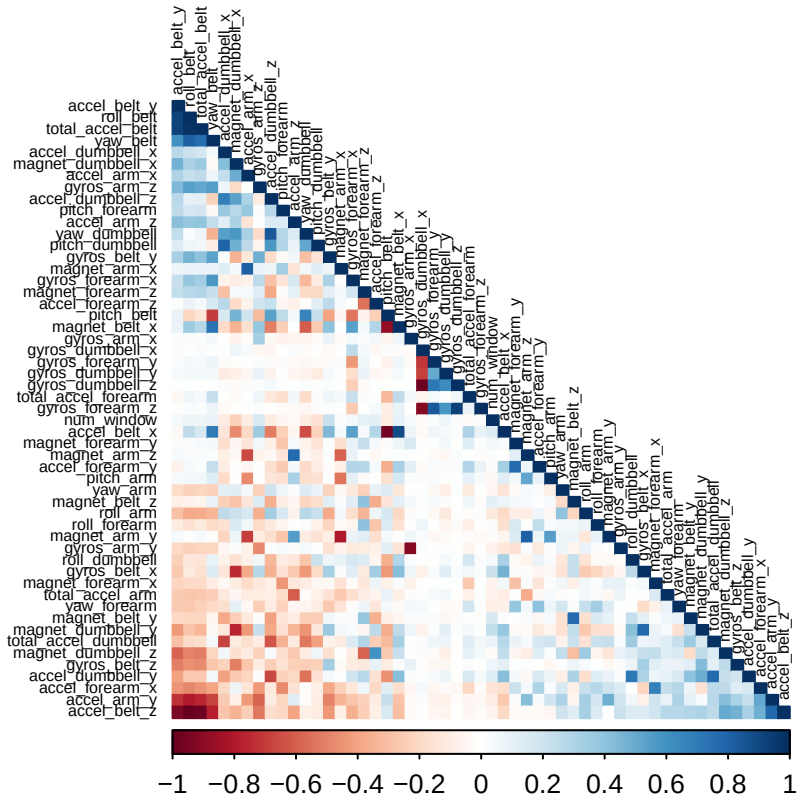
```
## [1] 19622   155
```

```
#Removing variables with Near Zero Variance
NZV <- nearZeroVar(train)
train <- train[, -NZV]
dim(train)
```

```
## [1] 19622    95
```

```
#Removing variables with mostly NA (90% threshold)
VarNA <- sapply(train, function(x) mean(is.na(x))) > 0.9
train <- train[, VarNA==FALSE]

#Analyzing if variables are correlated
corMat <- cor(train[, -54])
corrplot(corMat, order="FPC", method="color", type="lower",
         tl.cex=0.5, tl.col=rgb(0, 0, 0))
```




```
#High correlation between variables does not seem to be an issue. Variables will not be
#removed for this reason.
```

The training dataset was further divided into a training set (70%) and a validation set (30%). The testing dataset was only used to obtain the quiz results.

```
#Partition the training dataset
inTrain  <- createDataPartition(train$classe, p=0.7, list=FALSE)
training <- train[inTrain, ]
validating  <- train[-inTrain, ]

dim(training)
```

```
## [1] 13737    54
```

```
dim(validating)
```

```
## [1] 5885   54
```

Prediction Models

Three prediction models were used to find the one with the best accuracy. The models were built using the training data set. Those models were applied to the validating data set. The model with the best performance on the validatig data set was than used to make the predictions on the testing data set.

- Random Forest
- Decision Trees
- Generalized Boosted Model

Random Forest

```
set.seed(33633)

# Check for existing model file
modelRF <- "modelRF.RData"
if (!file.exists(modelRF)) {

    # If no file, set up parallel clusters
    require(parallel)
    require(doParallel)
    cl <- makeCluster(detectCores() - 1)
    registerDoParallel(cl)

    #Fit Random Forests model on training data
    RFCont <- trainControl(method="cv", number=3, verboseIter=FALSE)
    modelRF <- train(classe ~ ., data=training, method="rf", trControl=RFCont)

    save(modelRF, file = "modelRF.RData")

    stopCluster(cl)
} else {
    # Load model file if already exists
    load(file = "modelRF.RData", verbose = TRUE)
}
```

```
## Loading objects:
##   modelRF
```

```
modelRF$finalModel
```

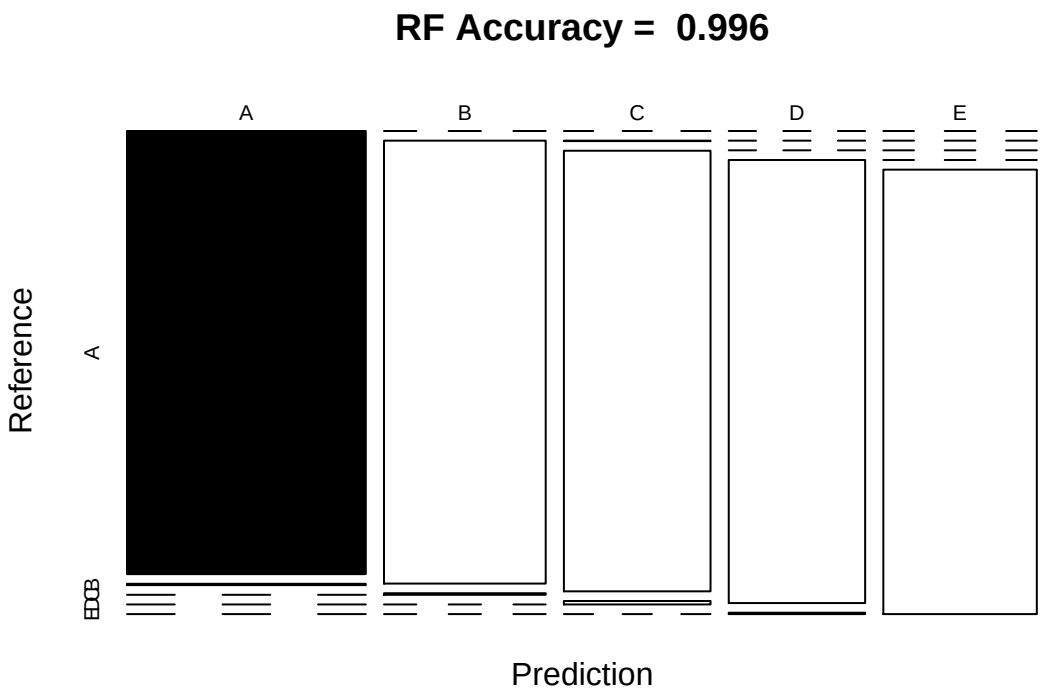
```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.24%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3904    1    0    0    1 0.0005120328
## B    6 2649    2    1    0 0.0033860045
## C    0    5 2389    2    0 0.0029215359
## D    0    0    9 2243    0 0.0039964476
## E    0    0    0    6 2519 0.0023762376
```

```
#Prediction on validating data set
predRF <- predict(modelRF, newdata=validating)
CM_RF <- confusionMatrix(predRF, validating$classe)
CM_RF
```

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

```
## Prediction    A    B    C    D    E
##          A 1674    5    0    0    0
##          B    0 1133    4    0    0
##          C    0    1 1022    8    0
##          D    0    0    0  956    3
##          E    0    0    0    0 1079
##
## Overall Statistics
##
##                Accuracy : 0.9964
##                  95% CI : (0.9946, 0.9978)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9955
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9947   0.9961   0.9917   0.9972
## Specificity            0.9988   0.9992   0.9981   0.9994   1.0000
## Pos Pred Value         0.9970   0.9965   0.9913   0.9969   1.0000
## Neg Pred Value         1.0000   0.9987   0.9992   0.9984   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1925   0.1737   0.1624   0.1833
## Detection Prevalence   0.2853   0.1932   0.1752   0.1630   0.1833
## Balanced Accuracy      0.9994   0.9969   0.9971   0.9955   0.9986
```

```r
#Confusion Matrix results plotted
plot(CM_RF$table, col=CM_RF$byClass,
     main=paste("RF Accuracy = ", round(CM_RF$overall['Accuracy'], 3)))
```

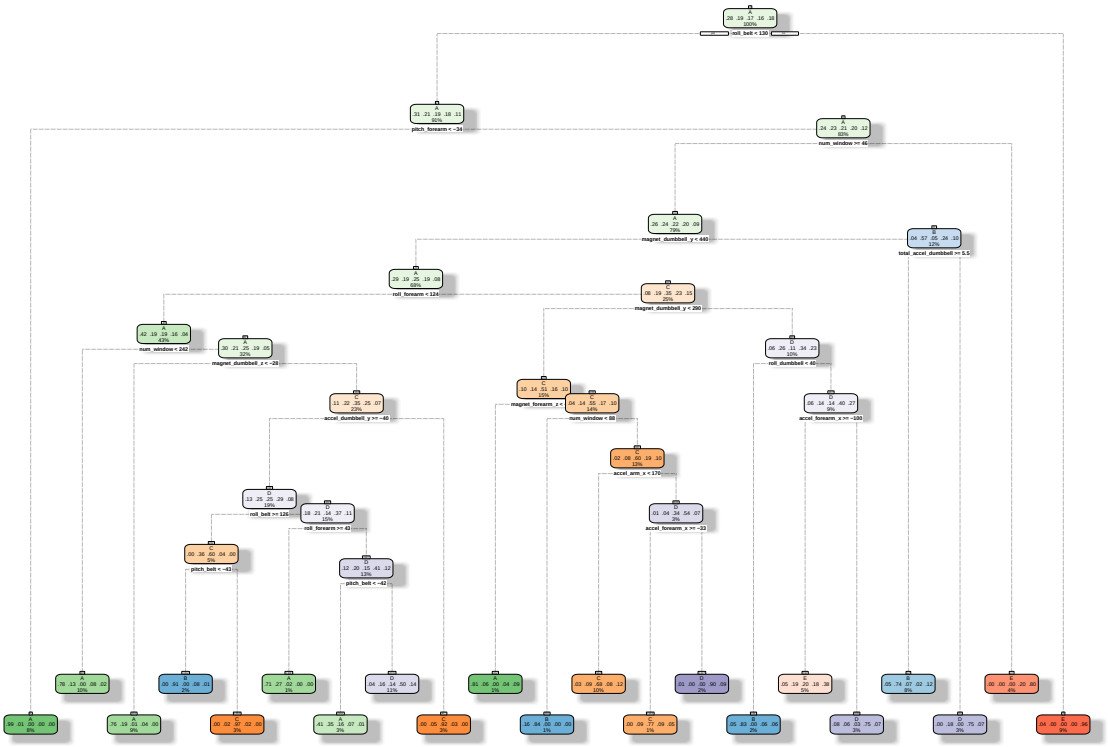


Decision Trees

```
set.seed(33633)

modelDT <- rpart(classe ~ ., data=training, method="class")

fancyRpartPlot(modelDT)
```



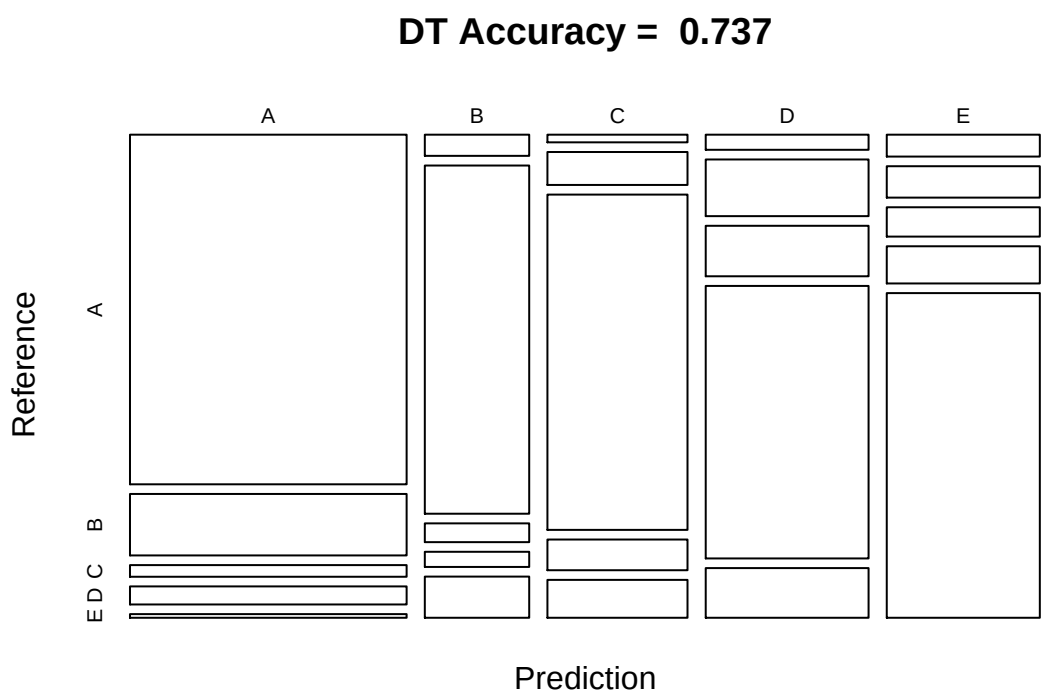
Rattle 2017−Oct−23 16:46:44 akovacs

```
predDT <- predict(modelDT, newdata=validating, type="class")
CM_DT <- confusionMatrix(predDT, validating$classe)
CM_DT
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1530  269   51   79   16
##          B   35  575   31   25   68
##          C   17   73  743   68   84
##          D   39  146  130  702  128
##          E   53   76   71   90  786
##
## Overall Statistics
##
##                Accuracy : 0.7368
##                  95% CI : (0.7253, 0.748)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6656
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9140  0.50483   0.7242   0.7282   0.7264
## Specificity            0.9014  0.96650   0.9502   0.9100   0.9396
## Pos Pred Value         0.7866  0.78338   0.7543   0.6131   0.7305
## Neg Pred Value         0.9635  0.89051   0.9422   0.9447   0.9384
## Prevalence             0.2845  0.19354   0.1743   0.1638   0.1839
## Detection Rate         0.2600  0.09771   0.1263   0.1193   0.1336
## Detection Prevalence   0.3305  0.12472   0.1674   0.1946   0.1828
## Balanced Accuracy      0.9077  0.73566   0.8372   0.8191   0.8330
```

```r
#Confusion Matrix results plotted
plot(CM_DT$table, col=CM_DT$byClass,
     main=paste("DT Accuracy = ", round(CM_DT$overall['Accuracy'], 3)))
```



Generalized Boosted Model

```r
set.seed(33633)


# Check for existing model file
modelGBM <- "modelGBM.RData"
if (!file.exists(modelGBM)) {

    # If no file, set up parallel clusters
    require(parallel)
```

```
    require(doParallel)
    cl <- makeCluster(detectCores() - 1)
    registerDoParallel(cl)

    #Fit GBM model on training data
    GBMCont <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
    modelGBM  <- train(classe ~ ., data=training, method = "gbm",
                       trControl = GBMCont, verbose = FALSE)

    save(modelGBM, file = "modelGBM.RData")

    stopCluster(cl)
} else {
    # Load model file if already exists
    load(file = "modelGBM.RData", verbose = TRUE)
}
```

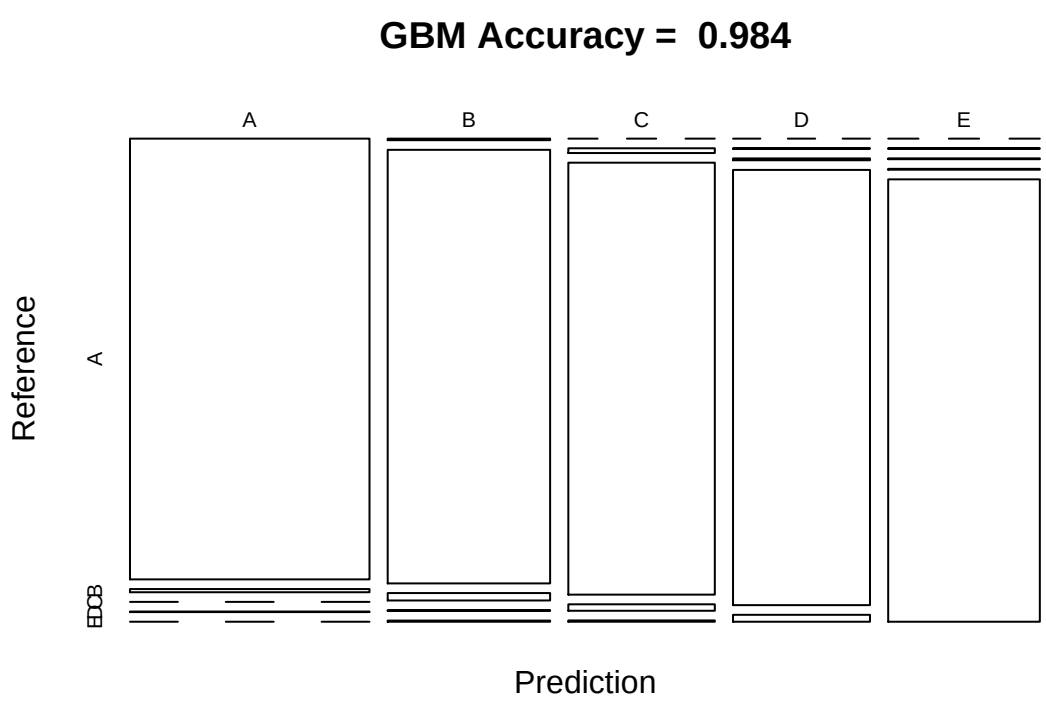
```
## Loading objects:
##   modelGBM
```

```
#Prediction on validating data set
predGBM <- predict(modelGBM, newdata=validating)
CM_GBM <- confusionMatrix(predGBM, validating$classe)
CM_GBM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1670   12    0    2    0
##          B    4 1114   19    2    3
##          C    0   11 1001   15    3
##          D    0    1    4  943   15
##          E    0    1    2    2 1061
##
## Overall Statistics
##
##                Accuracy : 0.9837
##                  95% CI : (0.9801, 0.9868)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9794
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9976   0.9781   0.9756   0.9782   0.9806
## Specificity            0.9967   0.9941   0.9940   0.9959   0.9990
## Pos Pred Value         0.9917   0.9755   0.9718   0.9792   0.9953
## Neg Pred Value         0.9990   0.9947   0.9949   0.9957   0.9956
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2838   0.1893   0.1701   0.1602   0.1803
## Detection Prevalence   0.2862   0.1941   0.1750   0.1636   0.1811
```

```
## Balanced Accuracy      0.9971   0.9861   0.9848   0.9871   0.9898
```

```
#Confusion Matrix results plotted
plot(CM_GBM$table, col=CM_GBM$byClass,
     main=paste("GBM Accuracy = ", round(CM_GBM$overall['Accuracy'], 3)))
```



Selected Model Applied to Test Data

Model Selection based on Accuracy

1. Random Forest:99.6%
2. Decision Trees: 73.7%
3. Generalized Boosted Model: 98.4%

The model with the highest accuracy is the Random Forest model. This model will be used to perform the prediction on the test data.

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

```
##  [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
```

The results from the testing data set were checked with the course project prediction quiz and yielded 100% correct answers.