Practical Machine Learning - Final Project

Elizabeth Storm

July 24, 2018

Exploratory Data Analysis

For this project we are using data from accelerometers from 6 participants. I am trying to predict whether a participant did an excerise in manner A or manner B, which is the classe variable on the record.

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har.

```
modeling <- read.csv('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv',
                     header=T, na.strings = c("NA","","#DIV/0!"))
holdout <- read.csv('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv',
                    header=T, na.strings = c("NA","","#DIV/0!"))
```

Second, I will preform a quick review of the data to get a handle on what the data looks like. I will also do any necessary steps to clean up missing data fields that would later on cause errors. I will also handle any highly correlated values.

```
head(modeling)
names(modeling)
```

```
# Remove unnecessary features
modeling<-modeling[,-1:-7]
# Remove features with no values
modeling<-modeling[, colSums(is.na(modeling)) < 1900]
# Check if there are other features that should be removed using nearZeroVar function
nearZero<-nearZeroVar(modeling, saveMetrics = TRUE)

#remove highly correlated variables
corr_data<-cor(modeling[,-length(modeling)])
correlated<-findCorrelation(corr_data, cutoff=0.8)
modeling<-modeling[, -correlated]
```

When I used the nearZero function, a review of the results showed that there were not any additional fields I would want to remove.

Data Prep for Modeling

I am going to partition my modeling data using a 60/40 split for training and testing. I am also going to do 5 fold cross validation when training my data.

```
set.seed(100)
inTrain = createDataPartition(modeling$classe, p = 0.6)[[1]]
train = modeling[ inTrain,]
test = modeling[-inTrain,]



fitControl <- trainControl(method = "cv",
```

```
                           number = 5,
                           allowParallel = FALSE)
```

Modeling

I am going to compare 4 types of models for classifying my data into the A/B types of exercise: - Random Forest (RF) - Gradiant Boosting (GBM) - Linear Discriminant Analysis (LDA) - A combination of the three above stacked as a Random Forest

```
set.seed(1000)
modelFitRF <- train(classe ~ ., method = 'rf', data=train, trControl=fitControl,
                    preprocess=c("center","scale"))
modelFitGMB <- train(classe ~ ., method = 'gbm', data=train, trControl=fitControl)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1103
##      2        1.5403             nan     0.1000    0.0826
##      3        1.4901             nan     0.1000    0.0578
##      4        1.4549             nan     0.1000    0.0502
##      5        1.4237             nan     0.1000    0.0449
##      6        1.3940             nan     0.1000    0.0413
##      7        1.3680             nan     0.1000    0.0319
##      8        1.3467             nan     0.1000    0.0324
##      9        1.3246             nan     0.1000    0.0324
##     10        1.3038             nan     0.1000    0.0261
##     20        1.1609             nan     0.1000    0.0166
##     40        0.9908             nan     0.1000    0.0091
##     60        0.8811             nan     0.1000    0.0039
##     80        0.8001             nan     0.1000    0.0048
##    100        0.7345             nan     0.1000    0.0041
##    120        0.6813             nan     0.1000    0.0026
##    140        0.6378             nan     0.1000    0.0015
##    150        0.6188             nan     0.1000    0.0024
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1647
##      2        1.5067             nan     0.1000    0.1165
##      3        1.4334             nan     0.1000    0.0928
##      4        1.3770             nan     0.1000    0.0691
##      5        1.3322             nan     0.1000    0.0685
##      6        1.2885             nan     0.1000    0.0589
##      7        1.2514             nan     0.1000    0.0620
##      8        1.2126             nan     0.1000    0.0515
##      9        1.1795             nan     0.1000    0.0414
##     10        1.1532             nan     0.1000    0.0356
##     20        0.9533             nan     0.1000    0.0184
##     40        0.7256             nan     0.1000    0.0125
##     60        0.5975             nan     0.1000    0.0129
##     80        0.5042             nan     0.1000    0.0036
##    100        0.4375             nan     0.1000    0.0031
##    120        0.3865             nan     0.1000    0.0026
##    140        0.3406             nan     0.1000    0.0013
##    150        0.3227             nan     0.1000    0.0013
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2086
##      2        1.4785             nan     0.1000    0.1549
##      3        1.3846             nan     0.1000    0.1155
##      4        1.3128             nan     0.1000    0.0908
##      5        1.2543             nan     0.1000    0.0814
##      6        1.2025             nan     0.1000    0.0735
##      7        1.1566             nan     0.1000    0.0722
##      8        1.1124             nan     0.1000    0.0533
##      9        1.0791             nan     0.1000    0.0512
##     10        1.0463             nan     0.1000    0.0431
##     20        0.8154             nan     0.1000    0.0238
##     40        0.5748             nan     0.1000    0.0114
##     60        0.4476             nan     0.1000    0.0068
##     80        0.3646             nan     0.1000    0.0027
##    100        0.3015             nan     0.1000    0.0037
##    120        0.2536             nan     0.1000    0.0032
##    140        0.2182             nan     0.1000    0.0023
##    150        0.2036             nan     0.1000    0.0011
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1161
##      2        1.5398             nan     0.1000    0.0779
##      3        1.4906             nan     0.1000    0.0618
##      4        1.4522             nan     0.1000    0.0468
##      5        1.4229             nan     0.1000    0.0395
##      6        1.3965             nan     0.1000    0.0401
##      7        1.3709             nan     0.1000    0.0384
##      8        1.3470             nan     0.1000    0.0309
##      9        1.3265             nan     0.1000    0.0286
##     10        1.3078             nan     0.1000    0.0282
##     20        1.1656             nan     0.1000    0.0183
##     40        0.9960             nan     0.1000    0.0079
##     60        0.8876             nan     0.1000    0.0063
##     80        0.8059             nan     0.1000    0.0050
##    100        0.7411             nan     0.1000    0.0043
##    120        0.6880             nan     0.1000    0.0027
##    140        0.6447             nan     0.1000    0.0015
##    150        0.6249             nan     0.1000    0.0017
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1698
##      2        1.5052             nan     0.1000    0.1158
##      3        1.4339             nan     0.1000    0.0945
##      4        1.3744             nan     0.1000    0.0719
##      5        1.3274             nan     0.1000    0.0681
##      6        1.2847             nan     0.1000    0.0619
##      7        1.2459             nan     0.1000    0.0527
##      8        1.2130             nan     0.1000    0.0531
##      9        1.1797             nan     0.1000    0.0396
##     10        1.1535             nan     0.1000    0.0409
##     20        0.9584             nan     0.1000    0.0206
##     40        0.7344             nan     0.1000    0.0115
```

```
##     60        0.6001             nan     0.1000    0.0091
##     80        0.5084             nan     0.1000    0.0055
##    100        0.4379             nan     0.1000    0.0043
##    120        0.3818             nan     0.1000    0.0015
##    140        0.3421             nan     0.1000    0.0017
##    150        0.3228             nan     0.1000    0.0031
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2148
##      2        1.4783             nan     0.1000    0.1499
##      3        1.3881             nan     0.1000    0.1154
##      4        1.3156             nan     0.1000    0.0983
##      5        1.2551             nan     0.1000    0.0809
##      6        1.2042             nan     0.1000    0.0634
##      7        1.1633             nan     0.1000    0.0636
##      8        1.1222             nan     0.1000    0.0543
##      9        1.0872             nan     0.1000    0.0512
##     10        1.0559             nan     0.1000    0.0434
##     20        0.8198             nan     0.1000    0.0238
##     40        0.5742             nan     0.1000    0.0077
##     60        0.4431             nan     0.1000    0.0098
##     80        0.3578             nan     0.1000    0.0045
##    100        0.2945             nan     0.1000    0.0024
##    120        0.2500             nan     0.1000    0.0019
##    140        0.2144             nan     0.1000    0.0015
##    150        0.2008             nan     0.1000    0.0012
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1069
##      2        1.5399             nan     0.1000    0.0790
##      3        1.4901             nan     0.1000    0.0611
##      4        1.4529             nan     0.1000    0.0462
##      5        1.4225             nan     0.1000    0.0421
##      6        1.3962             nan     0.1000    0.0390
##      7        1.3700             nan     0.1000    0.0389
##      8        1.3464             nan     0.1000    0.0303
##      9        1.3259             nan     0.1000    0.0259
##     10        1.3086             nan     0.1000    0.0282
##     20        1.1652             nan     0.1000    0.0162
##     40        0.9948             nan     0.1000    0.0094
##     60        0.8820             nan     0.1000    0.0070
##     80        0.8007             nan     0.1000    0.0046
##    100        0.7375             nan     0.1000    0.0031
##    120        0.6855             nan     0.1000    0.0028
##    140        0.6403             nan     0.1000    0.0016
##    150        0.6206             nan     0.1000    0.0029
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1643
##      2        1.5068             nan     0.1000    0.1195
##      3        1.4338             nan     0.1000    0.0873
##      4        1.3775             nan     0.1000    0.0827
##      5        1.3266             nan     0.1000    0.0650
##      6        1.2864             nan     0.1000    0.0656
```

```
##      7        1.2453             nan     0.1000    0.0487
##      8        1.2147             nan     0.1000    0.0502
##      9        1.1833             nan     0.1000    0.0416
##     10        1.1570             nan     0.1000    0.0333
##     20        0.9677             nan     0.1000    0.0196
##     40        0.7291             nan     0.1000    0.0100
##     60        0.5919             nan     0.1000    0.0068
##     80        0.5013             nan     0.1000    0.0055
##    100        0.4332             nan     0.1000    0.0048
##    120        0.3814             nan     0.1000    0.0031
##    140        0.3396             nan     0.1000    0.0024
##    150        0.3222             nan     0.1000    0.0018
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2180
##      2        1.4759             nan     0.1000    0.1480
##      3        1.3825             nan     0.1000    0.1146
##      4        1.3115             nan     0.1000    0.0962
##      5        1.2516             nan     0.1000    0.0804
##      6        1.2010             nan     0.1000    0.0617
##      7        1.1619             nan     0.1000    0.0663
##      8        1.1209             nan     0.1000    0.0481
##      9        1.0895             nan     0.1000    0.0568
##     10        1.0534             nan     0.1000    0.0401
##     20        0.8246             nan     0.1000    0.0225
##     40        0.5731             nan     0.1000    0.0124
##     60        0.4397             nan     0.1000    0.0051
##     80        0.3582             nan     0.1000    0.0044
##    100        0.2975             nan     0.1000    0.0042
##    120        0.2493             nan     0.1000    0.0015
##    140        0.2159             nan     0.1000    0.0021
##    150        0.2006             nan     0.1000    0.0007
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1128
##      2        1.5394             nan     0.1000    0.0783
##      3        1.4897             nan     0.1000    0.0612
##      4        1.4514             nan     0.1000    0.0478
##      5        1.4219             nan     0.1000    0.0438
##      6        1.3932             nan     0.1000    0.0362
##      7        1.3708             nan     0.1000    0.0424
##      8        1.3447             nan     0.1000    0.0305
##      9        1.3248             nan     0.1000    0.0264
##     10        1.3074             nan     0.1000    0.0295
##     20        1.1644             nan     0.1000    0.0172
##     40        0.9952             nan     0.1000    0.0094
##     60        0.8854             nan     0.1000    0.0054
##     80        0.8048             nan     0.1000    0.0046
##    100        0.7388             nan     0.1000    0.0036
##    120        0.6863             nan     0.1000    0.0031
##    140        0.6390             nan     0.1000    0.0011
##    150        0.6179             nan     0.1000    0.0016
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
```

```
##      1        1.6094             nan     0.1000    0.1651
##      2        1.5059             nan     0.1000    0.1191
##      3        1.4311             nan     0.1000    0.0999
##      4        1.3714             nan     0.1000    0.0751
##      5        1.3235             nan     0.1000    0.0624
##      6        1.2828             nan     0.1000    0.0696
##      7        1.2394             nan     0.1000    0.0500
##      8        1.2079             nan     0.1000    0.0454
##      9        1.1790             nan     0.1000    0.0462
##     10        1.1507             nan     0.1000    0.0376
##     20        0.9468             nan     0.1000    0.0188
##     40        0.7287             nan     0.1000    0.0111
##     60        0.6002             nan     0.1000    0.0074
##     80        0.5056             nan     0.1000    0.0036
##    100        0.4375             nan     0.1000    0.0034
##    120        0.3874             nan     0.1000    0.0026
##    140        0.3450             nan     0.1000    0.0047
##    150        0.3268             nan     0.1000    0.0018
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2184
##      2        1.4756             nan     0.1000    0.1454
##      3        1.3838             nan     0.1000    0.1151
##      4        1.3111             nan     0.1000    0.0929
##      5        1.2519             nan     0.1000    0.0773
##      6        1.2038             nan     0.1000    0.0717
##      7        1.1594             nan     0.1000    0.0543
##      8        1.1241             nan     0.1000    0.0609
##      9        1.0861             nan     0.1000    0.0597
##     10        1.0509             nan     0.1000    0.0383
##     20        0.8090             nan     0.1000    0.0243
##     40        0.5839             nan     0.1000    0.0145
##     60        0.4436             nan     0.1000    0.0062
##     80        0.3604             nan     0.1000    0.0036
##    100        0.2989             nan     0.1000    0.0026
##    120        0.2521             nan     0.1000    0.0020
##    140        0.2160             nan     0.1000    0.0017
##    150        0.2018             nan     0.1000    0.0013
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1090
##      2        1.5394             nan     0.1000    0.0752
##      3        1.4911             nan     0.1000    0.0590
##      4        1.4532             nan     0.1000    0.0497
##      5        1.4229             nan     0.1000    0.0397
##      6        1.3990             nan     0.1000    0.0351
##      7        1.3764             nan     0.1000    0.0406
##      8        1.3494             nan     0.1000    0.0314
##      9        1.3295             nan     0.1000    0.0327
##     10        1.3083             nan     0.1000    0.0290
##     20        1.1668             nan     0.1000    0.0152
##     40        0.9946             nan     0.1000    0.0088
##     60        0.8831             nan     0.1000    0.0056
##     80        0.8023             nan     0.1000    0.0041
```

```
##    100        0.7394             nan     0.1000    0.0036
##    120        0.6847             nan     0.1000    0.0036
##    140        0.6403             nan     0.1000    0.0018
##    150        0.6197             nan     0.1000    0.0017
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1651
##      2        1.5060             nan     0.1000    0.1179
##      3        1.4322             nan     0.1000    0.0957
##      4        1.3727             nan     0.1000    0.0775
##      5        1.3233             nan     0.1000    0.0628
##      6        1.2852             nan     0.1000    0.0618
##      7        1.2460             nan     0.1000    0.0567
##      8        1.2103             nan     0.1000    0.0486
##      9        1.1788             nan     0.1000    0.0495
##     10        1.1485             nan     0.1000    0.0379
##     20        0.9541             nan     0.1000    0.0213
##     40        0.7272             nan     0.1000    0.0073
##     60        0.5963             nan     0.1000    0.0066
##     80        0.4999             nan     0.1000    0.0046
##    100        0.4345             nan     0.1000    0.0023
##    120        0.3828             nan     0.1000    0.0023
##    140        0.3390             nan     0.1000    0.0017
##    150        0.3222             nan     0.1000    0.0015
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2168
##      2        1.4762             nan     0.1000    0.1460
##      3        1.3855             nan     0.1000    0.1099
##      4        1.3143             nan     0.1000    0.0937
##      5        1.2559             nan     0.1000    0.0761
##      6        1.2070             nan     0.1000    0.0650
##      7        1.1658             nan     0.1000    0.0564
##      8        1.1281             nan     0.1000    0.0632
##      9        1.0890             nan     0.1000    0.0536
##     10        1.0547             nan     0.1000    0.0401
##     20        0.8201             nan     0.1000    0.0183
##     40        0.5739             nan     0.1000    0.0127
##     60        0.4500             nan     0.1000    0.0074
##     80        0.3634             nan     0.1000    0.0061
##    100        0.2992             nan     0.1000    0.0032
##    120        0.2552             nan     0.1000    0.0026
##    140        0.2177             nan     0.1000    0.0016
##    150        0.2032             nan     0.1000    0.0021
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2157
##      2        1.4783             nan     0.1000    0.1504
##      3        1.3869             nan     0.1000    0.1093
##      4        1.3170             nan     0.1000    0.0954
##      5        1.2584             nan     0.1000    0.0805
##      6        1.2068             nan     0.1000    0.0710
##      7        1.1613             nan     0.1000    0.0704
##      8        1.1173             nan     0.1000    0.0517
```

```
##      9        1.0837             nan     0.1000    0.0454
##     10        1.0548             nan     0.1000    0.0532
##     20        0.8265             nan     0.1000    0.0327
##     40        0.5855             nan     0.1000    0.0119
##     60        0.4494             nan     0.1000    0.0073
##     80        0.3656             nan     0.1000    0.0052
##    100        0.3051             nan     0.1000    0.0018
##    120        0.2565             nan     0.1000    0.0014
##    140        0.2209             nan     0.1000    0.0019
##    150        0.2058             nan     0.1000    0.0012
```

```
modelFitLDA <- train(classe ~ ., method = 'lda', data=train, trControl=fitControl,
                     preprocess=c("center","scale"))


predRF <- predict(modelFitRF, newdata=test)
predGMB <- predict(modelFitGMB, newdata=test)
predLDA <- predict(modelFitLDA, newdata=test)


all_pred <- data.frame(predRF,predGMB,predLDA, classe = test$classe)
combinedMod <- train(classe ~ .,method="rf", data = all_pred, trControl=fitControl,
                     preprocess=c("center","scale"))
combinedPred <- predict(combinedMod, all_pred)
```

Evaluation of Results

Using accuarcy as a measurement to compare between model results, I will review how each model performs on my test data.

```
confusionMatrix(test$classe, predRF)$overall[1]
```

```
##  Accuracy
## 0.9914606
```

```
confusionMatrix(test$classe, predGMB)$overall[1]
```

```
##  Accuracy
## 0.9520775
```

```
confusionMatrix(test$classe, predLDA)$overall[1]
```

```
##  Accuracy
## 0.6417283
```

```
confusionMatrix(test$classe, combinedPred)$overall[1]
```

```
##  Accuracy
## 0.9914606
```

Based on the results, I see that Random Forest and the stacked model perform about the same. Since the Random Forest is simplier, I am going to rely on that model to assess my performance on the Holdout data as part of our Course Quiz.

```
holdoutRF <- predict(modelFitRF, newdata=holdout)

holdoutRF
```

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

These are the values I submitted to the quiz which resulted in 19/20 accuracy which is good!