Practical Machine Learning Course Project

Code

Joshua Paolo Acilo

Introduction

This work is an attempt to predict the manner in which the people did the exercise in the following study. *Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.*

In the study mentioned above, six participants participated in a dumbell lifting activity five different ways. The five ways, as described in the study, were "exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes."

By processing the data gathered from accelerometers on the belt, forearm, arm, and dumbell of the participants in a machine learning algorithm, can the appropriate activity (class A-E) be predicted?

This report underlines the following task discussed in coursera. 1. How you built your model. 2. How you used cross validation. 3. What you think the expected out of sample error is 4. Why you made the choices you did

---

Data Pre Processing

Load the following libraries

Set a seed for reproducibility of this study

Hide

```
set.seed(42)
```

Load the train and test data

Hide

```
train <- fread('pml-training.csv')
train$V1 = NULL
train = as.data.frame(train)
test  <- fread('pml-testing.csv')
test$V1 = NULL
test = as.data.frame(test)
```

CHECK

Hide

```
c(dim(train),dim(test))
```

```
[1] 19622   159    20   159
```

Split the training set into 70-30

Hide

```
train_part  <- createDataPartition(train$classe, p=0.70, list=FALSE)
train_train <- train[train_part,]
train_valid  <- train[-train_part,]
```

CHECK

Hide

```
c(dim(train_train),dim(train_valid))
```

```
[1] 13737   159  5885   159
```

Feature Selection

Drop the features with near zero variance

Hide

```
nzv <- nearZeroVar(train_train)
train_train <- train_train[,-nzv]
train_valid  <- train_valid[,-nzv]
```

The features with near zero variance are features that doesn't show much variation and therefore is less important in contributing to make a correct prediction as it cannot discern the variability among the targets.

CHECK

Hide

```
c(dim(train_train),dim(train_valid))
```

```
[1] 13737   129  5885   129
```

Drop the features that are mostly NA using 95 threshold

Hide

```
all_na <- sapply(train_train, function(x) mean(is.na(x))) > 0.95
train_train <- train_train[,all_na==FALSE]
train_valid  <- train_valid[,all_na==FALSE]
```

The features with >=95% missing are dropped as these features does not give the model much information in order to make a correct prediction.

CHECK

Hide

```
c(dim(train_train),dim(train_valid))
```

```
[1] 13737    58  5885    58
```

Drop the identifier features

Hide

```
train_train <- train_train[,-(1:5)]
train_valid  <- train_valid[,-(1:5)]
```

The identifier features are directly correlated to each of the targets, and is therefore needed to be dropped.

CHECK

Hide

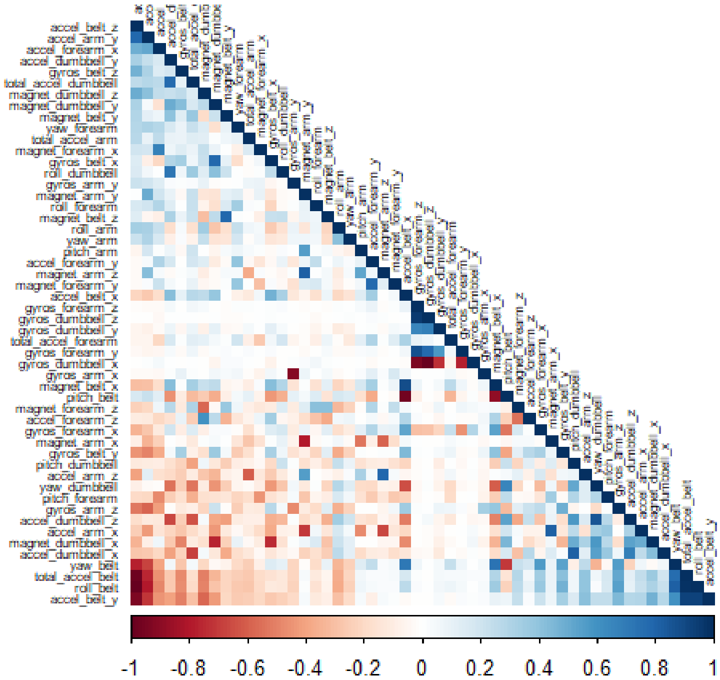
```
c(dim(train_train),dim(train_valid))
```

```
[1] 13737    53  5885    53
```

Perform correlation analysis

Hide

```
corr_mat <- cor(train_train[, -53])
corrplot(corr_mat,order="FPC",method="color",type="lower",
         tl.cex = 0.5, tl.col = rgb(0, 0, 0))
```



The colors in the plot above shows the strength of correlation among pairs of features. The darker the color is, the more correlated the pair of features are (red - negatively correlated, blue - positively correlated). Since the strong correlations are just few, further reduction of the number of features is not explored.

Model Building

Set a seed for reproducibility of this study

Hide

```
set.seed(42)
```

RF Model Building 1

Hide

```
RF <- trainControl(method="cv",number=5,verboseIter=FALSE)
RF <- train(classe~.,data=train_train,method="rf",trControl=RF)
RF$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: 0.69%
Confusion matrix:
     A    B    C    D    E  class.error
A 3903    2    0    0    1 0.0007680492
B   21 2629    8    0    0 0.0109104590
C    0   11 2374   11    0 0.0091819699
D    0    1   25 2224    2 0.0124333925
E    0    1    6    6 2512 0.0051485149
```

RF Model Building 2

Hide

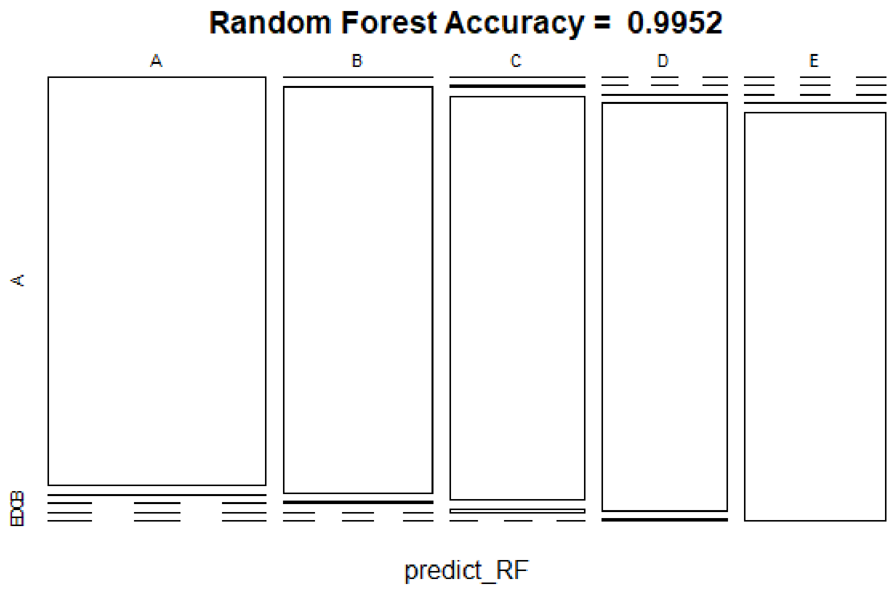
```
predict_RF <- predict(RF, newdata=train_valid)
confusion_RF <- confusionMatrix(table(predict_RF, train_valid$classe))
confusion_RF
```

```
Confusion Matrix and Statistics


predict_RF    A    B    C    D    E
         A 1670    1    0    0    0
         B    3 1135    4    0    0
         C    1    3 1021    9    0
         D    0    0    1  954    5
         E    0    0    0    1 1077

Overall Statistics

               Accuracy : 0.9952
                 95% CI : (0.9931, 0.9968)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.994

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9976   0.9965   0.9951   0.9896   0.9954
Specificity            0.9998   0.9985   0.9973   0.9988   0.9998
Pos Pred Value         0.9994   0.9939   0.9874   0.9937   0.9991
Neg Pred Value         0.9991   0.9992   0.9990   0.9980   0.9990
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2838   0.1929   0.1735   0.1621   0.1830
Detection Prevalence   0.2839   0.1941   0.1757   0.1631   0.1832
Balanced Accuracy      0.9987   0.9975   0.9962   0.9942   0.9976
```

RF Model Building 3

Hide

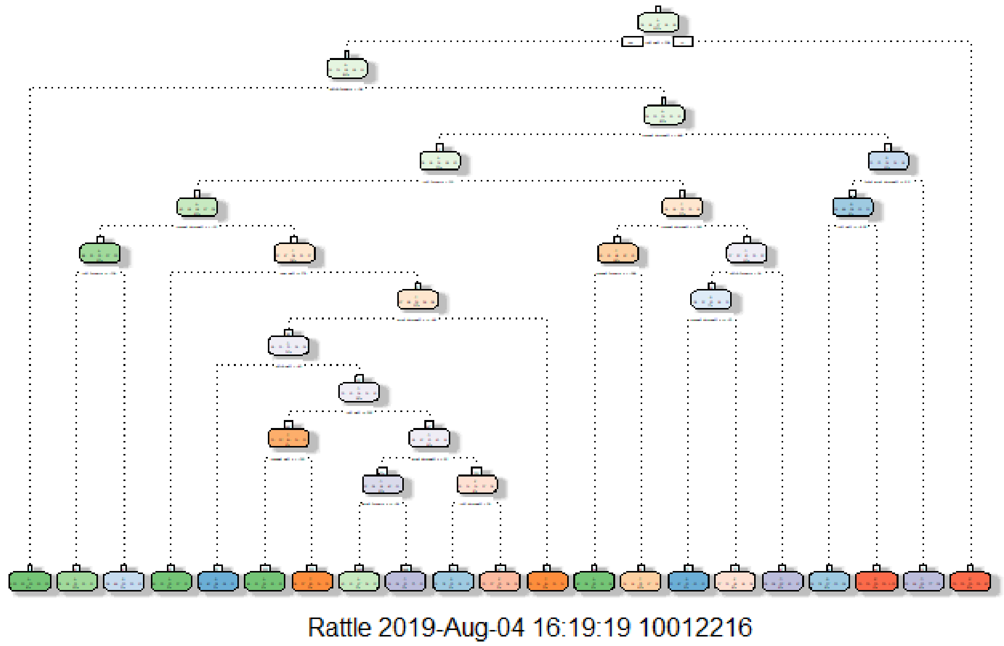
```
plot(confusion_RF$table, col = confusion_RF$byClass,
     main = paste("Random Forest Accuracy = ",round(confusion_RF$overall['Accuracy'], 4)))
```



DT Model Building 1

Hide

```r
DT <- rpart(classe~.,data=train_valid,method="class")
fancyRpartPlot(DT)
```



DT Model Building 2

Hide

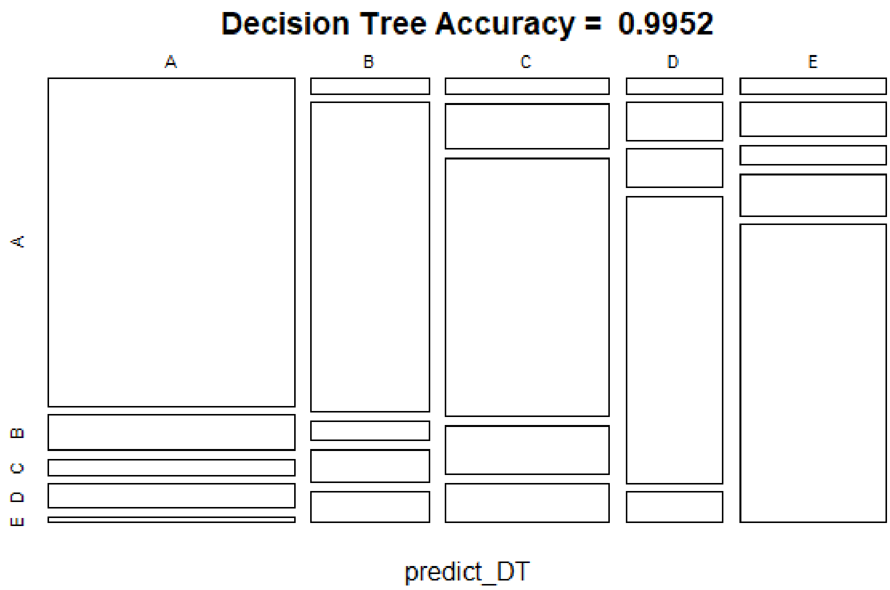
```r
predict_DT <- predict(DT,newdata=train_valid,type="class")
confusion_DT <- confusionMatrix(table(predict_DT,train_valid$classe))
confusion_DT
```

```
Confusion Matrix and Statistics


predict_DT    A    B    C    D    E
         A 1512  164   72  111   26
         B   36  676   43   72   69
         C   53  139  787  149  119
         D   29   69   70  519   56
         E   44   91   54  113  812

Overall Statistics

               Accuracy : 0.7317
                 95% CI : (0.7202, 0.743)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6591

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9032   0.5935   0.7671  0.53838   0.7505
Specificity            0.9114   0.9536   0.9053  0.95448   0.9371
Pos Pred Value         0.8021   0.7545   0.6311  0.69852   0.7289
Neg Pred Value         0.9595   0.9072   0.9485  0.91346   0.9434
Prevalence             0.2845   0.1935   0.1743  0.16381   0.1839
Detection Rate         0.2569   0.1149   0.1337  0.08819   0.1380
Detection Prevalence   0.3203   0.1523   0.2119  0.12625   0.1893
Balanced Accuracy      0.9073   0.7736   0.8362  0.74643   0.8438
```

DT Model Building 3

Hide

```
plot(confusion_DT$table,col=confusion_DT$byClass,
     main = paste("Decision Tree Accuracy = ",round(confusion_RF$overall['Accuracy'], 4)))
```



GBM Model Building 1

Hide

```
GBM <- trainControl(method="repeatedcv",number=5,repeats=1)
GBM  <- train(classe~.,data=train_valid,method="gbm",trControl=GBM,verbose=FALSE)
GBM$finalModel
```

```
A gradient boosted model with multinomial loss function.
150 iterations were performed.
There were 52 predictors of which 52 had non-zero influence.
```

GBM Model Building 2

Hide

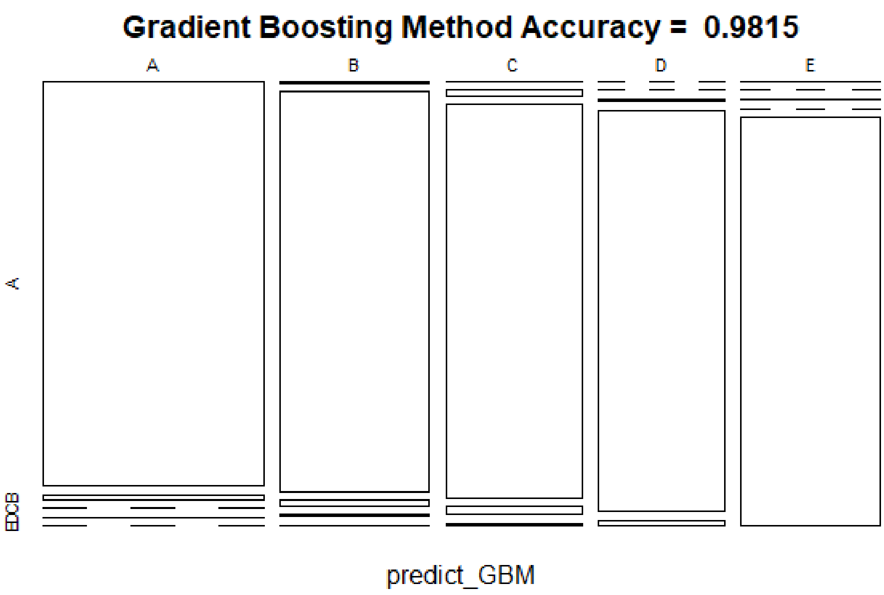
```
predict_GBM <- predict(GBM,newdata=train_valid)
confusion_GBM <- confusionMatrix(table(predict_GBM,train_valid$classe))
confusion_GBM
```

```
Confusion Matrix and Statistics


predict_GBM    A    B    C    D    E
          A 1667   18    0    1    0
          B    5 1108   16    3    4
          C    1   13 1003   20    6
          D    0    0    6  940   14
          E    1    0    1    0 1058

Overall Statistics

               Accuracy : 0.9815
                 95% CI : (0.9777, 0.9848)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9766

 Mcnemar's Test P-Value : 5.471e-06

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9958   0.9728   0.9776   0.9751   0.9778
Specificity            0.9955   0.9941   0.9918   0.9959   0.9996
Pos Pred Value         0.9887   0.9754   0.9616   0.9792   0.9981
Neg Pred Value         0.9983   0.9935   0.9952   0.9951   0.9950
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2833   0.1883   0.1704   0.1597   0.1798
Detection Prevalence   0.2865   0.1930   0.1772   0.1631   0.1801
Balanced Accuracy      0.9957   0.9834   0.9847   0.9855   0.9887
```

GBM Model Building 3

Hide

```
plot(confusion_GBM$table,col=confusion_GBM$byClass,
     main = paste("Gradient Boosting Method Accuracy = ",round(confusion_GBM$overall['Accuracy'], 4)))
```



Predict the test set using the RF

Hide

```
predict_TEST <- predict(RF,newdata=test)
predict_TEST[1:20]
```

```
 [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
```

Predict the test set using the DT

Hide

```
predict_TEST <- predict(DT,newdata=test)
predict_TEST[1:20]
```

```
 [1] 0.03608847 0.78833107 0.06764706 0.03571429 0.42724458 0.03608847 0.07569721 0.03571429
 [9] 0.99591837 0.78833107 0.03608847 0.06764706 0.03608847 0.99591837 0.07569721 0.03086420
[17] 0.98529412 0.09375000 0.09375000 0.02985075
```

Predict the test set using the GBM

Hide

```
predict_TEST <- predict(GBM,newdata=test)
predict_TEST[1:20]
```

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

Conclusion

Based on the metrics presented above, the Random Forest (RF) performed the best in the prediction task. A total of 500 trees were used by the RF with an accuracy of 99.46% in the validation set. Five fold cross validation is performed in the model building in order to get a more accurate measurement of the performance of the trained model. This model is applied in the holdout dataset and the first 20 predictions is presented above.