How Well Exercises Are Done

Hoi Leung

June 2, 2016

The main goal for this report is to export how well exercises are being done. Model performance will be based on the accuracy statistics.

Initial Data Exploration

```
library(caret)
set.seed(1234)
#download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
,
#              destfile = "pml-training.csv")
#download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
#              destfile = "pml-testing.csv")
pml_train_full <- read.csv("pml-training.csv")
#pml_test <- read.csv("pml-testing.csv")
dim(pml_train_full)
```

```
## [1] 19622   160
```

Because the number of columns and rows of the data, only half of the rows are used for the analysis to improve run time.

```
train_ind <- sample(1:dim(pml_train_full)[1], dim(pml_train_full)[1]/2, replace=F)
pml_train <- pml_train_full[train_ind, ]
pml_test <- pml_train_full[-train_ind, ]
dim(pml_train)
```

```
## [1] 9811  160
```

```
table(pml_train$classe)
```

```
##
##    A    B    C    D    E
## 2772 1951 1697 1596 1795
```

```
#str(pml_train)
```

Upon first inspection, many variables can be converted from string to numbers. However, given the amount of time available to do this analysis, all the non-numeric variables are simply removed. Furthermore, missing values are replaced with the median values.

```
numeric_var <- sapply(pml_train[, 1:160],is.numeric)
#sum(numeric_var)
numeric_var[1:7] <- FALSE
numeric_var[160] <- TRUE
pml_train <- pml_train[, numeric_var]
pml_test <- pml_test[, numeric_var]

mi <- preProcess(pml_train[, -120], method="medianImpute")
pml_train <- predict(mi, newdata=pml_train)
pml_test <- predict(mi, newdata=pml_test)
```

Model Training Data

Tree, random forest, and GBM were be used to predict classe. Afterward, another tree is built with the outputs of the previous 3 methods as inputs. Except for random forest, all the models are run with default options. For random forest, to shorten run time, only 100 trees were built.

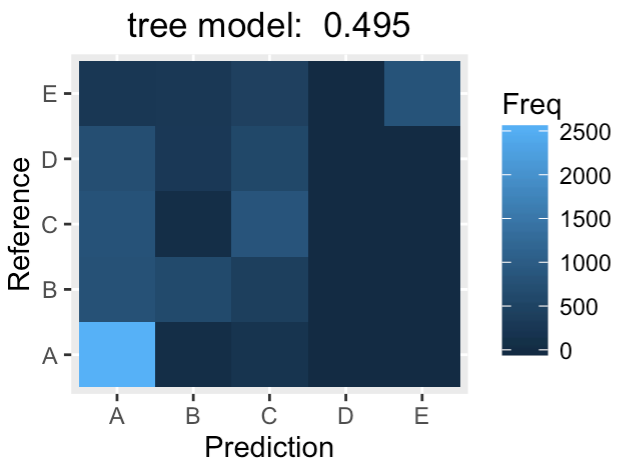
```
#gbm_fit <- train(classe ~ ., data=pml_train, method="gbm")
#rf_fit <- train(classe ~ ., data=pml_train, method="rf", ntree = 100)
#tree_fit <- train(classe ~ ., data=pml_train, method="rpart")
#save(gbm_fit, file="gbm_fit.RData")
#save(rf_fit, file="rf_fit.RData")
#save(tree_fit, file="tree_fit.RData")
load("gbm_fit.RData"); load("rf_fit.RData"); load("tree_fit.RData")
pml_train_comb <- data.frame(gbm_pred=gbm_fit$train$.outcome,
                             rf_pred=rf_fit$train$.outcome,
                             tree_pred=tree_fit$train$.outcome,
                             classe=pml_train$classe)
comb_tree_fit <- train(classe ~ ., data=pml_train_comb, method="rpart")

acc <-
rbind(confusionMatrix(pml_train_comb$gbm_pred, pml_train_comb$classe)$overall[1],
      confusionMatrix(pml_train_comb$rf_pred, pml_train_comb$classe)$overall[1],
      confusionMatrix(pml_train_comb$tree_pred, pml_train_comb$classe)$overall[1],
      confusionMatrix(comb_tree_fit$train$.outcome, pml_train_comb$classe)$overall[1]
)
acc
```

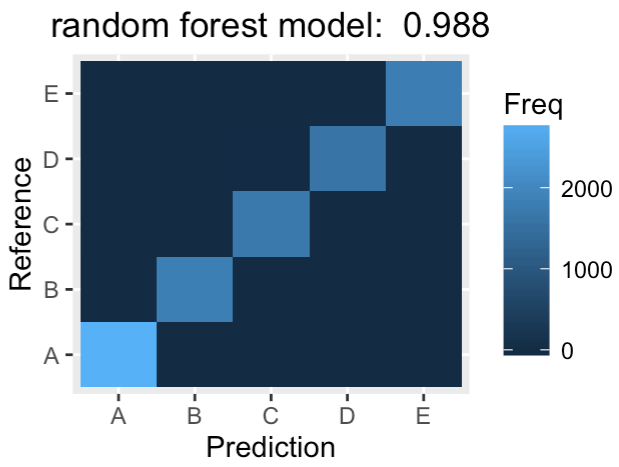
```
##      Accuracy
## [1,]        1
## [2,]        1
## [3,]        1
## [4,]        1
```

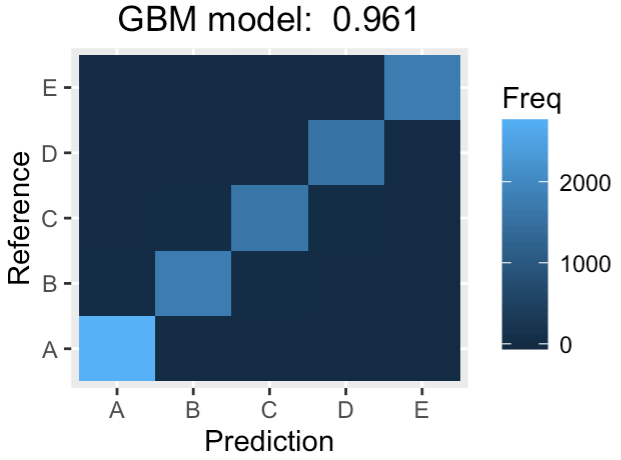
Examine Accuracy on Test Data

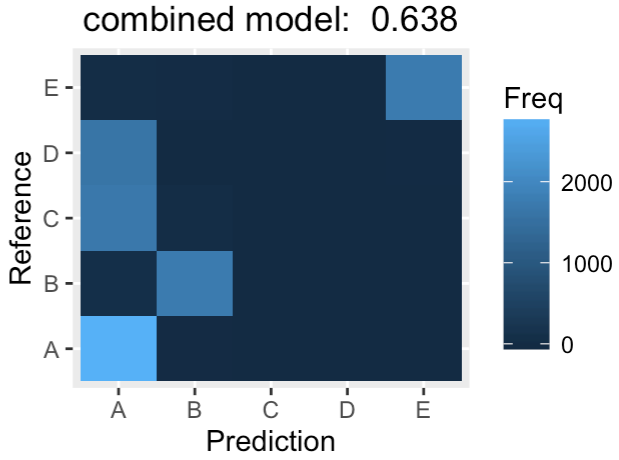
The test data were scored with the 4 models. Afterward, the heat maps and their accuracy statistics are plotted.

```
gbm_pred <- predict(gbm_fit, newdata=pml_test)
rf_pred <- predict(rf_fit, newdata=pml_test)
tree_pred <- predict(tree_fit, newdata=pml_test)
pml_test_comb <- data.frame(gbm_pred=gbm_pred,
                            rf_pred=rf_pred,
                            tree_pred=tree_pred,
                            classe=pml_test$classe)
comb_tree_pred <- predict(comb_tree_fit, newdata=pml_test_comb)

gbm_test_accuracy <- confusionMatrix(gbm_pred, pml_test$classe)$overall[1]
rf_test_accuracy <- confusionMatrix(rf_pred, pml_test$classe)$overall[1]
tree_test_accuracy <- confusionMatrix(tree_pred, pml_test$classe)$overall[1]
combine_test_accuracy <- confusionMatrix(comb_tree_pred, pml_test$classe)$overall[1]

tree_table <- confusionMatrix(tree_pred, pml_test$classe)$table
rf_table <- confusionMatrix(rf_pred, pml_test$classe)$table
gbm_table <- confusionMatrix(gbm_pred, pml_test$classe)$table
comb_table <- confusionMatrix(comb_tree_pred, pml_test$classe)$table

library(grid)
library(gridExtra)
g1 <- ggplot(data.frame(tree_table))
g1 <- g1 + geom_tile(aes(x=Prediction, y=Reference, fill=Freq))
g1 <- g1 + ggtitle(paste("tree model: ", round(tree_test_accuracy, 3)))

g2 <- ggplot(data.frame(rf_table))
g2 <- g2 + geom_tile(aes(x=Prediction, y=Reference, fill=Freq))
g2 <- g2 + ggtitle(paste("random forest model: ", round(rf_test_accuracy, 3)))

g3 <- ggplot(data.frame(gbm_table))
g3 <- g3 + geom_tile(aes(x=Prediction, y=Reference, fill=Freq))
g3 <- g3 + ggtitle(paste("GBM model: ", round(gbm_test_accuracy, 3)))

g4 <- ggplot(data.frame(comb_table))
g4 <- g4 + geom_tile(aes(x=Prediction, y=Reference, fill=Freq))
g4 <- g4 + ggtitle(paste("combined model: ", round(combine_test_accuracy, 3)))

grid.arrange(g1, g2, g3, g4, ncol = 2, nrow=2)
```









Conclusion

The random forest model happened to have the highest accuracy statistics in the test sample. Therefore, it will be selected as the final model.