Prediction Assignment Writeup - ML

Shawn Paul

```
> library(lattice)
> library(ggplot2)
> library(caret)
> library(rpart)
> library(rpart.plot)
> library(corrplot)
corrplot 0.90 loaded
> library(rattle)
Loading required package: tibble
Loading required package: bitops
Rattle: A free graphical interface for data science with R.
Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
>
>
> library(RColorBrewer)
>
> set.seed(222)
> url_train <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
> url_quiz  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
>
> data_train <- read.csv(url(url_train), strip.white = TRUE, na.strings = c("NA",""))
> data_quiz  <- read.csv(url(url_quiz),  strip.white = TRUE, na.strings = c("NA",""))
>
> dim(data_train)
[1] 19622   160
```

Create 2 partitions (75% & 25%) within training set

```
> dim(train_set)
[1] 14718   160
> dim(test_set)
[1] 4904  160
>
```

Remove NA values and near-zero variance variables, both to be removed together.

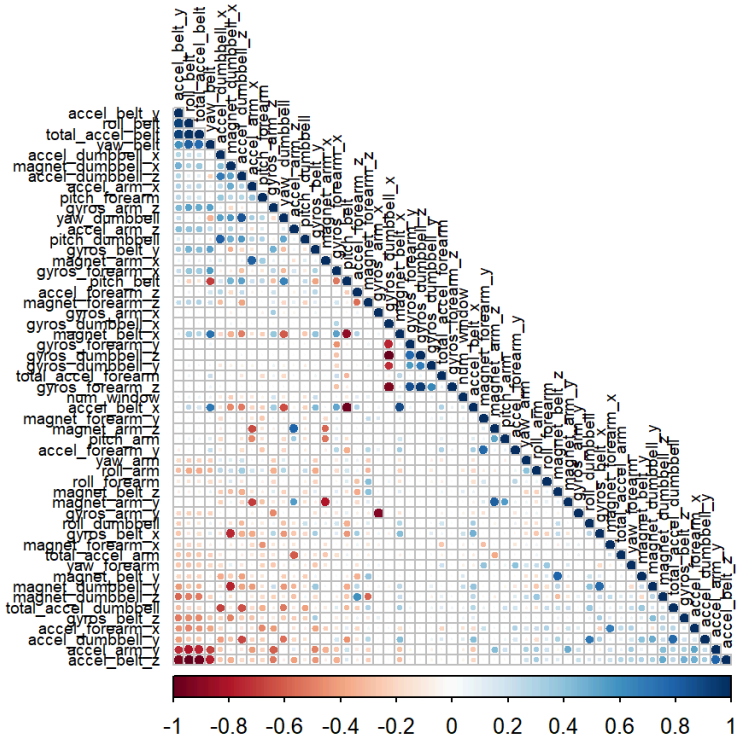
```
> nzv_var <- nearZeroVar(train_set)
>
> train_set <- train_set[ , -nzv_var]
> test_set  <- test_set [ , -nzv_var]
>
> dim(train_set)
[1] 14718   120
```

Remove variables that are mostly NA, a threshold of 95% is selected.

```
> na_var <- sapply(train_set, function(x) mean(is.na(x))) > 0.95
> train_set <- train_set[ , na_var == FALSE]
> test_set  <- test_set [ , na_var == FALSE]
>
> dim(train_set)
[1] 14718    59
> dim(test_set)
[1] 4904   59
>
```

Correlation Analysis

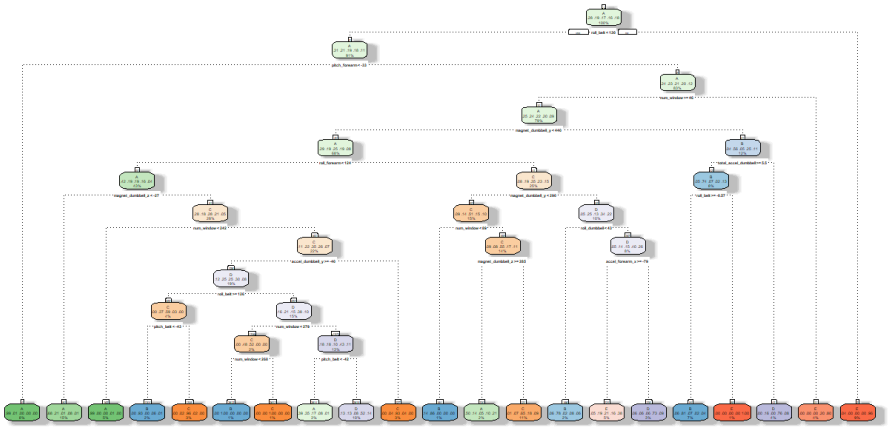
```
> corr_matrix <- cor(train_set[ , -54])
> corrplot(corr_matrix, order = "FPC", method = "circle", type = "lower",
+          tl.cex = 0.6, tl.col = rgb(0, 0, 0))
```



color shows the correlations; the darker blue showing a positive correlation and the darker red showing a negative correlation. Due to so few strong correlations, a few prediction models will be built for better accuracy.

Prediction Models: Decision Tree Model

```
> set.seed(2222)
> fit_decision_tree <- rpart(classe ~ ., data = train_set, method="class")
> fancyRpartPlot(fit_decision_tree)
```



Rattle 2022-Apr-13 09:42:47 mllrg

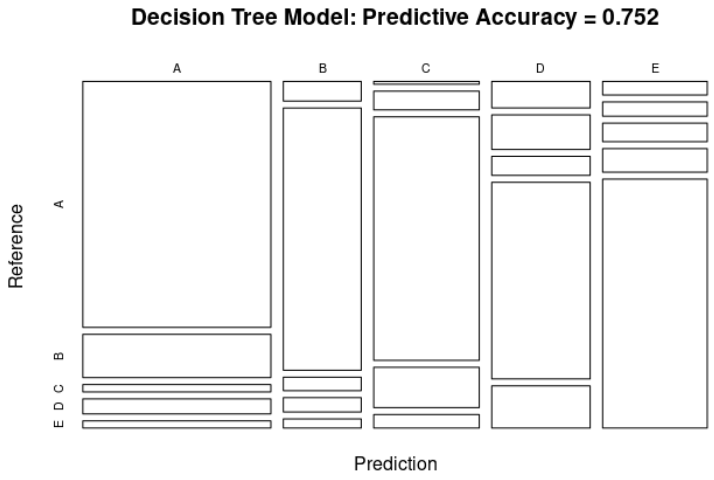
Predictions of the decision tree model with test_set

```
> predict_decision_tree <- predict(fit_decision_tree, newdata = test_set, type="class")
> conf_matrix_decision_tree <- confusionMatrix(predict_decision_tree, factor(test_set$classe))
> conf_matrix_decision_tree
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1238  218   37   76   36
         B   41  547   28   30   19
         C    8   53  688  114   38
         D   70   91   50  518  111
         E   38   40   52   66  697

Overall Statistics

               Accuracy : 0.752
                 95% CI : (0.7397, 0.7641)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.685

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

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.8875   0.5764   0.8047   0.6443   0.7736
Specificity            0.8954   0.9702   0.9474   0.9215   0.9510
Pos Pred Value         0.7713   0.8226   0.7636   0.6167   0.7805
Neg Pred Value         0.9524   0.9052   0.9583   0.9296   0.9491
Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
Detection Rate         0.2524   0.1115   0.1403   0.1056   0.1421
Detection Prevalence   0.3273   0.1356   0.1837   0.1713   0.1821
Balanced Accuracy      0.8914   0.7733   0.8760   0.7829   0.8623
```

The predictive accuracy of the decision tree model is relatively low at 75.2 %.

Plot the predictive accuracy of the decision tree model.

```
> plot(conf_matrix_decision_tree$table, col = conf_matrix_decision_tree$byClass,
+      main = paste("Decision Tree Model: Predictive Accuracy =",
+                   round(conf_matrix_decision_tree$overall['Accuracy'], 4)))
```



Generalized Boosted Model (GBM)

```
> set.seed(2222)
> ctrl_GBM <- trainControl(method = "repeatedcv", number = 5, repeats = 2)
> fit_GBM  <- train(classe ~ ., data = train_set, method = "gbm",
+                   trControl = ctrl_GBM, verbose = FALSE)

> fit_GBM$finalModel
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1392    5    0    1    0
##          B    3  931    4    1    5
##          C    0   12  843    9    2
##          D    0    1    8  789   10
##          E    0    0    0    4  884
##
## Overall Statistics
##
##                Accuracy : 0.9867
##                  95% CI : (0.9831, 0.9898)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9832
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9978   0.9810   0.9860   0.9813   0.9811
## Specificity            0.9983   0.9967   0.9943   0.9954   0.9990
## Pos Pred Value         0.9957   0.9862   0.9734   0.9765   0.9955
## Neg Pred Value         0.9991   0.9955   0.9970   0.9963   0.9958
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2838   0.1898   0.1719   0.1609   0.1803
## Detection Prevalence   0.2851   0.1925   0.1766   0.1648   0.1811
## Balanced Accuracy      0.9981   0.9889   0.9901   0.9884   0.9901
```