Prediction of Proper Workout Form based on Wearable Data

```
## Warning: package 'gbm' was built under R version 3.5.2
```

Executive Summary

Data from the Weight Lifting Exercise Dataset by Velloso et al. was analyzed in order to determine if it was possible to predict whether exercises were performed correctly.Training data was imported into R and pre-processed to remove missing data, near-zero variance variables and correlated predictors. The data was split into training and test sets and several classification models were evaluated to determine ability to predict on the data. The final model was used to predict exercise quality on a separate test set, one without assigned classes to the exercises, which was processed in a similar manner.

Data Processing

Training data was downloaded from the URL link. The data contained 160 possible predictors for 19622 samples. Inspecting the data set revealed the first 7 predictors related to participant name and time the exercise was performed. Since the person performing the activity and the time the activity was performed likely have little to do with whether or not an exercise was perfomed correctly, these first 7 predictors were removed. Further inspection of the dataset revealed that several predictors contained significant missing values. Since many models have issues with missing values, those predictors with greater than 80% missing values were removed from the training set.

This left 86 variables on which to predict outcomes. Further processing of the data was then conducted to remove predictors with near zero variance as these can skew model perofrmance. In addition, highly correlated predictors were removed in order to improve performance.

The processed data set contained 46 predictors. Finally, the data set was split into testing and training sets in order to train the prediction model and estimate the out-of-sample accuracy of the model. The training data contained 75% of the samples, and the test set 25%.

Model Building

Two classification models were built in order to compare performance. Since the goal of the prediction was a high level of accuracy, and interpretation of the results was less important, random forest and boosted tree (GBM) models were selected. These models can provide high

accuracy and are adept at muli-class classification problems.

Repeated k-fold cross-validation was chosen in order to tune the model parameters. Due to the large number of samples, 5 folds and 3 repetitions was selected for cross-validation using the below code:

```
trainCtrl <- trainControl(method = "repeatedcv",
                          number = 5,
                          repeats = 3,
                          verboseIter = TRUE)
```

The default values for tuning parameters was used for the random forest model. For the GBM model, a grid of values for the tuning parameters was used in order to determine the most optimal fit:

```
grid <- expand.grid(n.trees=c(50, 150, 250, 500),
                    shrinkage=c(0.05, 0.1, 0.3),
                    n.minobsinnode = c(1, 5, 10),
                    interaction.depth=c(1, 5, 10))
```

Models were then trained on the sub-training set (containing 75% of the samples from the total training set).

```
rf_mod <- train(classe ~., data = training, method = "rf", trControl = trainCtrl)
gbm_mod <- train(classe ~., data = training, method = "gbm", trControl = trainCtr
l, tuneGrid = grid)
```

Plots of the model tuning showed that 150 trees, 0.1 shrinkage, 10 minimum observations in a node and an interaction depth of 7 would get high accuracy for the GBM model while not requiring too much computational time. R automatically selected 23 for the parameter *mtry* as the optimal tuning parameter for the random forest model.

Model evaluation

After model building and tuning, the out-of-sample error of the models were evaluated using the sub-testing set (containing 25% of the total training set).

```
predRF <- predict(rf_mod, testing)
predGBM <- predict(gbm_mod, testing)
```

```
##   model  Accuracy     Kappa AccuracyLower AccuracyUpper AccuracyNull
## 1   GBM 0.9942904 0.9927768     0.9917585     0.9962027    0.2844617
## 2    RF 0.9877651 0.9845221     0.9842790     0.9906508    0.2844617
##   AccuracyPValue McnemarPValue
## 1              0           NaN
## 2              0           NaN
```

The random forest model demonstrates superior performance, so this was selected as the final model:

```
##    case prediction
## 1     1          B
## 2     2          A
## 3     3          B
## 4     4          A
## 5     5          A
## 6     6          E
## 7     7          D
## 8     8          B
## 9     9          A
## 10   10          A
## 11   11          B
## 12   12          C
## 13   13          B
## 14   14          A
## 15   15          E
## 16   16          E
## 17   17          A
## 18   18          B
## 19   19          B
## 20   20          B
```