# PracticalMachineLearningCourseProject

Raul Rodrigo Olivarez Jr

2023-08-05

## Overview

This serves as the concluding report for the Practical Machine Learning course on Coursera, which is part of the Data Science Specialization track provided by John Hopkins University.

The objective of this project involves utilizing data collected from accelerometers placed on various body parts (belt, forearm, arm, and dumbell) of six participants. The goal is to predict the manner in which these individuals performed their exercises, which is indicated by the "classe" variable within the training dataset.

To achieve this, we employ four distinct models: Decision Tree, Random Forest, and Support Vector Machine. These models are trained using Weka_control as cross-validation on the training dataset. Subsequently, we apply these trained models to a validation dataset, which is randomly extracted from the training CSV data. This step allows us to calculate accuracy. By analyzing these metrics, we determine the optimal model. Once identified, we employ this model to predict outcomes for 20 cases using the test CSV dataset.

## Libraries and Set Seed for Reproducibility

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(RWeka)
library(lattice)
library(ggplot2)

library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
set.seed(1234)
```

### Load the Dataset

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainingcsv <- read.csv(url(trainUrl), na.strings=c("NA","#DIV/0!",""),stringsAsFactors = TRUE)
testingcsv <- read.csv(url(testUrl), na.strings=c("NA","#DIV/0!",""),stringsAsFactors=TRUE)
```

### Cleaning the Data

Removing unnecessary variables (N/A) and columns which are irrelevant when creating the model

```
trainingcsv <- trainingcsv[,colMeans(is.na(trainingcsv)) < .9]
trainingcsv <- trainingcsv[,-c(1:7)]
dim(trainingcsv)
```

```
## [1] 19622    53
```

Splitting the training dataset into sub-training and validation sets. The testing csv file will be utilized for the quiz cases.

```
inTrain <- createDataPartition(y=trainingcsv$classe, p=0.7, list=F)
training <- trainingcsv[inTrain,]
validation <- trainingcsv[-inTrain,]
```

## Creating and Testing Models

We will be using some of the popular models such as Random Forest, SVM, Decision Trees and Gradient Boosted Trees and compare them which of them best fits.

### Random Forest

```
RF <- make_Weka_classifier("weka/classifiers/trees/RandomForest")
rfmodel <- RF(classe~ ., data=training, control = Weka_control(K=floor(2*sqrt(18))))
predrfmodel <- predict(rfmodel, validation)
cmrf <- confusionMatrix(predrfmodel, factor(validation$classe))
cmrf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    0    0    1    0
##          B    1 1136   12    0    0
##          C    0    3 1013    7    0
##          D    0    0    1  955    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
##                Accuracy : 0.9956
##                  95% CI : (0.9935, 0.9971)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9944
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9974   0.9873   0.9907   1.0000
## Specificity            0.9998   0.9973   0.9979   0.9998   0.9998
## Pos Pred Value         0.9994   0.9887   0.9902   0.9990   0.9991
## Neg Pred Value         0.9998   0.9994   0.9973   0.9982   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1930   0.1721   0.1623   0.1839
## Detection Prevalence   0.2845   0.1952   0.1738   0.1624   0.1840
## Balanced Accuracy      0.9996   0.9973   0.9926   0.9952   0.9999
```

## Support Vector Machine

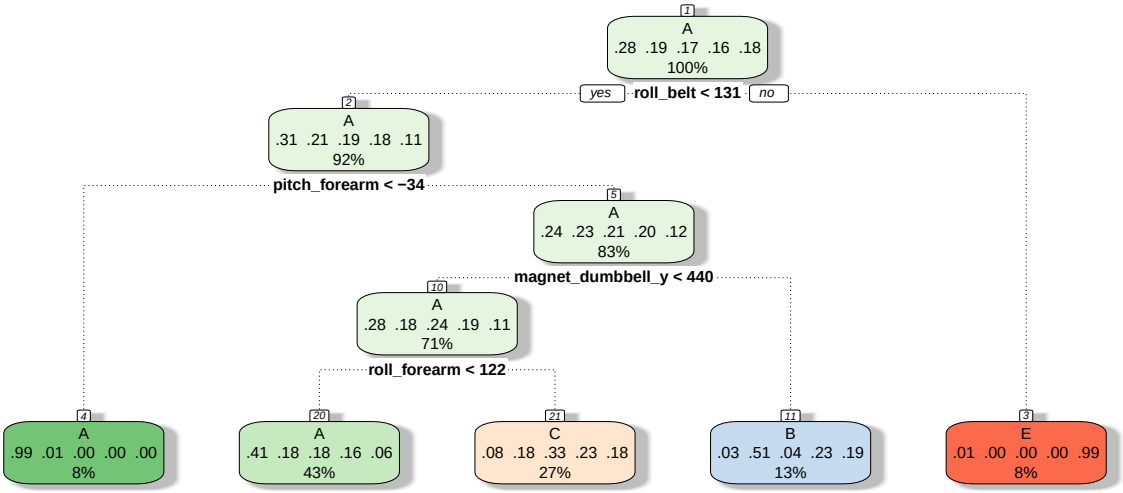
```
SVMModel <- SMO(classe ~., data=training, control = Weka_control(C='1', K = list("PolyKernel", E = 2)))
predSVMModel <- predict(SVMModel, validation)
cmSVM <- confusionMatrix(predSVMModel, factor(validation$classe))
cmSVM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
```

```
##          A 1667   35    2    3    0
##          B    4 1071   27    0   10
##          C    2   19  985   70   13
##          D    1    2    5  891   10
##          E    0   12    7    0 1049
##
## Overall Statistics
##
##                Accuracy : 0.9623
##                  95% CI : (0.9571, 0.967)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9522
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9958   0.9403   0.9600   0.9243   0.9695
## Specificity            0.9905   0.9914   0.9786   0.9963   0.9960
## Pos Pred Value         0.9766   0.9631   0.9045   0.9802   0.9822
## Neg Pred Value         0.9983   0.9858   0.9915   0.9853   0.9931
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2833   0.1820   0.1674   0.1514   0.1782
## Detection Prevalence   0.2901   0.1890   0.1850   0.1545   0.1815
## Balanced Accuracy      0.9932   0.9658   0.9693   0.9603   0.9828
```

## Decision Trees

```
DecisionTreeModel <- train(classe ~ .,method="rpart",data=training)
fancyRpartPlot(DecisionTreeModel$finalModel)
```




Rattle 2023−Aug−05 18:51:01 raulrodrigoolivarezjr.

```
predDecisionTreeModel <- predict(DecisionTreeModel, validation)
cmdt <- confusionMatrix(predDecisionTreeModel, factor(validation$classe))
cmdt
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1519  473  484  451  156
##          B   28  401   45  167  148
##          C  123  265  497  346  289
##          D    0    0    0    0    0
##          E    4    0    0    0  489
##
## Overall Statistics
##
##                Accuracy : 0.4938
##                  95% CI : (0.4809, 0.5067)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.338
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9074  0.35206  0.48441   0.0000  0.45194
## Specificity            0.6286  0.91825  0.78946   1.0000  0.99917
## Pos Pred Value         0.4927  0.50824  0.32697      NaN  0.99189
## Neg Pred Value         0.9447  0.85518  0.87881   0.8362  0.89002
## Prevalence             0.2845  0.19354  0.17434   0.1638  0.18386
## Detection Rate         0.2581  0.06814  0.08445   0.0000  0.08309
## Detection Prevalence   0.5239  0.13407  0.25828   0.0000  0.08377
## Balanced Accuracy      0.7680  0.63516  0.63693   0.5000  0.72555
```

Among the three, the best model is the Random Forest, with 99.56% accuracy that we can estimate to be our out-of-sample error. When the model was fitted using the training data, it lead to a 100% accuracy which we can assume as our in-sample error. Thus, this is a good model to use for our test sets.

**Random Forest using Training**

```
predrfmodeltr <- predict(rfmodel, training)
cmrftr <- confusionMatrix(predrfmodeltr, factor(training$classe))
cmrftr
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

**Random Forest using Validation**

```
predrfmodel <- predict(rfmodel, validation)
cmrf <- confusionMatrix(predrfmodel, factor(validation$classe))
cmrf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    0    0    1    0
##          B    1 1136   12    0    0
##          C    0    3 1013    7    0
##          D    0    0    1  955    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
##                Accuracy : 0.9956
##                  95% CI : (0.9935, 0.9971)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9944
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9974   0.9873   0.9907   1.0000
## Specificity            0.9998   0.9973   0.9979   0.9998   0.9998
## Pos Pred Value         0.9994   0.9887   0.9902   0.9990   0.9991
## Neg Pred Value         0.9998   0.9994   0.9973   0.9982   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1930   0.1721   0.1623   0.1839
## Detection Prevalence   0.2845   0.1952   0.1738   0.1624   0.1840
## Balanced Accuracy      0.9996   0.9973   0.9926   0.9952   0.9999
```

# Final Prediction for the Test

Executing the Random Forest model on our test set to forecast the outcome of the "classe" variable (with 5 possible levels) for a total of 20 cases.

```
predRFtest <- predict(rfmodel, testingcsv)
print(predRFtest)
```

```
##  [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
```

## Generating Files for Quiz Submission

```
pml_write_files = function(x) {
  n = length(x)
  for (i in 1:n) {
    filename = paste0("problem_id_", i, ".txt")
    write.table(x[i], file=filename, quote=FALSE,row.names=FALSE, col.names=FALSE)
  }
}

pml_write_files(predRFtest)
```