Practical Machine Learning - Final Project

Sandeep Rangarajen

`{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)`

Overview

This is the final report for Coursera's Practical Machine Learning course, as part of the Data Science Specialization track offered by John Hopkins.

In this project, we will use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise. This is the "classe" variable in the training set. We train 4 models: **Decision Tree**, **Random Forest**, **Gradient Boosted Trees**, **Support Vector Machine** using k-folds cross validation on the training set. We then predict using a validation set randomly selected from the training csv data to obtain the **accuracy** and **out of sample error rate**. Based on those numbers, we decide on the best model, and use it to predict 20 cases using the test csv set. (Data Links are Below - Local versions: Downloaded June 16/2020)

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Loading Data and Libraries

Loading all the libraries and the data `{r results='hide', message=FALSE} library(lattice) library(ggplot2) library(caret) library(kernlab) library(rattle) library(corrplot) set.seed(1234)`

```
traincsv <- read.csv("./data/pml-training.csv")
testcsv <- read.csv("./data/pml-testing.csv")
dim(traincsv)
dim(testcsv)
```

#We see that there are 160 variables and 19622 observations in the training set, while 20 for the test set.

Cleaning the Data

#Removing unnecessary variables. Starting with N/A variables.

```
traincsv <- traincsv[,colMeans(is.na(traincsv)) < .9] #removing mostly na columns
traincsv <- traincsv[,-c(1:7)] #removing metadata which is irrelevant to the outcome
```

#Removing near zero variance variables. `{r nzv} nvz <- nearZeroVar(traincsv) traincsv <- traincsv[,-nvz] dim(traincsv)`

#Now that we have finished removing the unnecessary variables, we can now split the training set into a **validation** and sub **training** set. The testing set "testcsv" will be left alone, and used for the final quiz test cases.

```
inTrain <- createDataPartition(y=traincsv$classe, p=0.7, list=F)
train <- traincsv[inTrain,]
valid <- traincsv[-inTrain,]
```

Creating and Testing the Models

##Here we will test a few popular models including: **Decision Trees**, **Random Forest**, **Gradient Boosted Trees**, and **SVM**. This is probably more than we will need to test, but just out of curiosity and good practice we will run them for comparison.

##Set up control for training to use 3-fold cross validation.

```
control <- trainControl(method="cv", number=3, verboseIter=F)
```

Decision Tree

##**Model:**

`{r, cache=TRUE} mod_trees <- train(classe~., data=train, method="rpart", trControl = control, tuneLength = 5) fancyRpartPlot(mod_trees$finalModel)`

##**Prediction:**

```
pred_trees <- predict(mod_trees, valid)
cmtrees <- confusionMatrix(pred_trees, factor(valid$classe))
cmtrees
```

Random Forest

`{r, cache=TRUE} mod_rf <- train(classe~., data=train, method="rf", trControl = control, tuneLength = 5) pred_rf <- predict(mod_rf, valid) cmrf <- confusionMatrix(pred_rf, factor(valid$classe) cmrf`

Gradient Boosted Trees

```
{r, cache=TRUE} mod_gbm <- train(classe~., data=train, method="gbm", trControl = control,
tuneLength = 5, verbose = F) pred_gbm <- predict(mod_gbm, valid) cmgbm <- confusionMatrix(pred_gbm,
factor(valid$classe)) cmgbm
```

Support Vector Machine

```
{r, cache=TRUE} mod_svm <- train(classe~., data=train, method="svmLinear", trControl
= control, tuneLength = 5, verbose = F) pred_svm <- predict(mod_svm, valid) cmsvm <-
confusionMatrix(pred_svm, factor(valid$classe)) cmsvm
```

Results (Accuracy & Out of Sample Error)

```
{r, echo=FALSE} models <- c("Tree", "RF", "GBM", "SVM") accuracy <- round(c( cmtrees$overall[1],
cmrf$overall[1], cmgbm$overall[1], cmsvm$overall[1]),3) #accuracy oos_error <- 1 - accuracy
#out of sample error data.frame(accuracy = accuracy, oos_error = oos_error, row.names =
models)
```

##The best model is the Random Forest model, with `r cmrf$overall[1]` accuracy and `r 1-cmrf$overall[1]` out of sample error rate. We find that to be a sufficient enough model to use for our test sets. **

Predictions on Test Set

Running our test set to predict the classe (5 levels) outcome for 20 cases with the Random Forest model.

```
{r, echo=FALSE} pred <- predict(mod_rf, testcsv) print(pred)
```

Appendix

##correlation matrix of variables in training set `{r, echo=FALSE} corrPlot <- cor(train[, -length(names(train))]) corrplot(corrPlot, method="color")`

##Plotting the models `{r, echo=FALSE} plot(mod_trees) plot(mod_rf) plot(mod_gbm)`