Course 8 project

Niamh Devitt

Analysis of the quality of exercise taken, using data from a number of different exercise recording devices

Executive Summary

This report explores not what kind of activity is taken but how well it is taken. To do this I've used data on 6 participants who were asked to do a series of exercises correctly and incorrectly.

Data Cleaning and Exploration

First we download the training and test datasets

```
library(caret)
```

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

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

```
library(rpart)
url_train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
url_test <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(url_train, destfile = "train_data.csv")
download.file(url_test, destfile = "test_data.csv")

train_data <- read.csv("~/DataScience/Course8_PracticalMachineLearning/train_data.csv")
test_data <- read.csv("~/DataScience/Course8_PracticalMachineLearning/test_data.csv")
```

Data cleaning; we remove a number of variables with near zero variance that will have little impact in our prediction. First 7 columns are not predictors and we'll remove columns that are mostly blank.

```
train_clean <- train_data[,-c(1:7)]
remove_blanks <- which(colSums(is.na(train_clean) |train_clean=="")>0.9*dim(train_clean)[1])
train_clean <- train_clean[,-remove_blanks]

test_clean <- test_data[,-c(1,7)]
remove_blanks2 <- which(colSums(is.na(test_clean) |test_clean=="")>0.9*dim(test_clean)[1])
test_clean <- test_clean[,-remove_blanks2]
```

First we will split the training data into a training dataset and a validation set where we will test our final model before applying to the test set and submitting.

```
set.seed(383)
inTrain <- createDataPartition(train_clean$classe, p=0.7, list= FALSE)
training <- train_clean[inTrain,]
validation <- train_clean[-inTrain,]

dim(training)
```

```
## [1] 13737    53
```

```
dim(validation)
```

```
## [1] 5885   53
```

```
levels(training$classe)
```

```
## [1] "A" "B" "C" "D" "E"
```

In order to avoid overfitting and to assess the effectiveness of our models we will use cross-validation.

Model 1: Classification Tree

```
set.seed(111)
mod_cv <- trainControl(method="cv", number=3, verboseIter=FALSE)
modfit_ct<- train(classe~., data=training, method="rpart", trControl=mod_cv)
pred_ct <- predict(modfit_ct, newdata=validation)
confusionMatrix(pred_ct, validation$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.4948173
```

The accuracy of our classification tree is low at 49.5%. We'll fit a random forest model to improve accuracy.

Model 2: Random Forest

```
set.seed(456)
mod_cv <- trainControl(method="cv", number=3, verboseIter=FALSE)
modfit_rf <- train(classe ~., data = training, method = "rf", trControl = mod_cv)
modfit_rf$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.79%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3901    3    1    0    1 0.001280082
## B   21 2627    9    1    0 0.011662904
## C    0   11 2373   12    0 0.009599332
## D    0    2   30 2217    3 0.015541741
## E    0    1    5    8 2511 0.005544554
```

```
pred_rf <- predict(modfit_rf, newdata=validation)
confusionMatrix(pred_rf, validation$classe)$overall['Accuracy']
```

```
## Accuracy
## 0.993373
```

Random forest model prediction; our random forest model has 99.3% accuracy with cross validation 3 times in predicting the validation set.

This is a very strong model. but we will try a gradient boosting model also to see what kind of accuracy we find.

Model 3: Gradient Boosting

The gradient boosting model has 96% accuracy with cross validation.

```
set.seed(444)
modfit_gbm <- train(classe ~., data = training, method = "gbm", verbose=FALSE, trControl = mod_cv)
modfit_gbm$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 52 had non-zero influence.
```

```
pred_gbm <- predict(modfit_gbm, newdata=validation)
confusionMatrix(pred_gbm, validation$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.9612574
```

Model 3: Combined

```
combined <- data.frame(pred_ct, pred_rf, pred_gbm, classe=validation$classe)
combined_rf <- train(classe~., data = combined, method = "rf")
pred_combinedrf <- predict(combined_rf, newdata = validation)
confusionMatrix(pred_combinedrf, validation$classe)$overall['Accuracy']
```

```
## Accuracy
## 0.993373
```

We can try to stack our predictions to improve accuracy, however we can see that accuracy has not improved from the random foret model so we will use this model as it will have less bias and easier to interpret.

Our final model, the random forest model has 99.2% accuracy, expected out of sample error will be less than 1%.

Using our Final Model to predict on the Clean Test set

```
FinalTest_pred <- predict(modfit_rf,newdata=test_clean)
FinalTest_pred
```

```
##  [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
```

The results wll be tested in Course Project Quiz.