Project_WriteUp

Alex Humfrey

13 August 2020

Data Loading and Pre Processing

Necessary libraries were uploaded, Data was downloaded and loaded into R. Variables which add no value to the model and variables with only NA values were removed from the data set before the model creation stage.

```
library(dplyr); library(ggplot2);library(caret); library(randomForest); library(rpart)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rpart.plot); library(RColorBrewer); library(rattle); library(gbm); library(
corrplot)
```

```
## 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.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
## Loaded gbm 2.1.8
```

```
## corrplot 0.84 loaded
```

```
url_train <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
url_test <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

destfile_train <- "/Users/alexhumfrey/Documents/Data Science Coursera/Scripts/Pract
icalMachineLearning/train_data.csv"
destfile_test <- "/Users/alexhumfrey/Documents/Data Science Coursera/Scripts/Practi
calMachineLearning/test_data.csv"

download.file(url_train, destfile = destfile_train)
download.file(url_test, destfile = destfile_test)

training <- read.csv(destfile_train)
dim(training)
```

```
## [1] 19622   160
```

```
testing <- read.csv(destfile_test)
dim(testing)
```

```
## [1]  20 160
```

There are 19622 obs. of 160 variables in the training set, and there are 20 obs of 160 variables in the test set. Lots of variables have only NAs.

```
# remove variables w/ missing vales
train_data <- training[, colSums(is.na(training)) == 0]
test_data <- testing[, colSums(is.na(testing)) == 0]

# remove variables which have no impact on the classe of the exercise
train_data <- train_data[, -(1:7)]
test_data  <- test_data[, -(1:7)]

dim(train_data);dim(test_data)
```

```
## [1] 19622    86
```

```
## [1] 20 53
```

Prediction Preparation and Exploratory Analysis

The training data set was split into a validation set and a training set, a correlation plot was created to show the strength of linear correlation between each of the variables in the model. We can see from the correlation plot that there aren't too many variables that are highly correlated therefore it was decided that a PCA was not necessary.

```
# split to training  and validation sets
set.seed(12432)
inTrain <- createDataPartition(train_data$classe, p = 0.7, list = FALSE)
train_data <- train_data[inTrain, ]
valid_data <- train_data[-inTrain, ]
dim(train_data)
```

```
## [1] 13737    86
```

```
dim(valid_data)
```

```
## [1] 4110   86
```

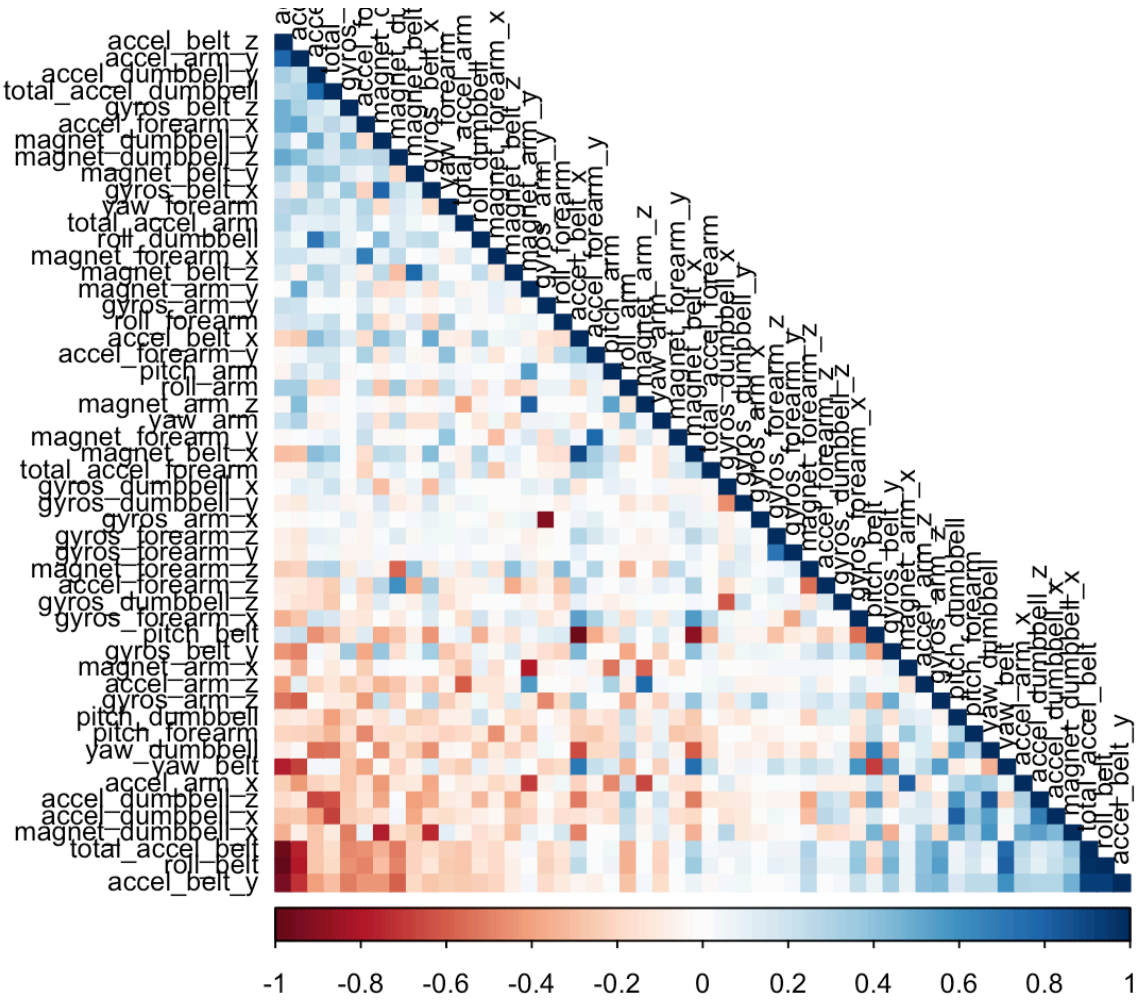
```
#remove variables with near zero variance
NZV <- nearZeroVar(train_data)
train_data <- train_data[, -NZV]
valid_data  <- valid_data[, -NZV]
dim(train_data)
```

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

```
dim(valid_data)
```

```
## [1] 4110   53
```

```
# predictor correlation plots
corMatrix <- cor(train_data[, -53])
corrplot(corMatrix, order = "FPC", method = "color", type = "lower",
         tl.cex = 0.8, tl.col = rgb(0, 0, 0))
```



Model Creation and Prediction on Validation Set

Three different methods were used to create prediction models for the data. 1. Random Forest model. 2. Classification Tree Model. 3. Generalised Boosting Model. A summary of each of the models is provided.

```
# random forest - model creation, model summary
set.seed(125)
RFModel <- randomForest(classe ~ ., data=train_data, ntree = 500, importance = TRUE
)
RFModel$finalModel
```

```
## NULL
```

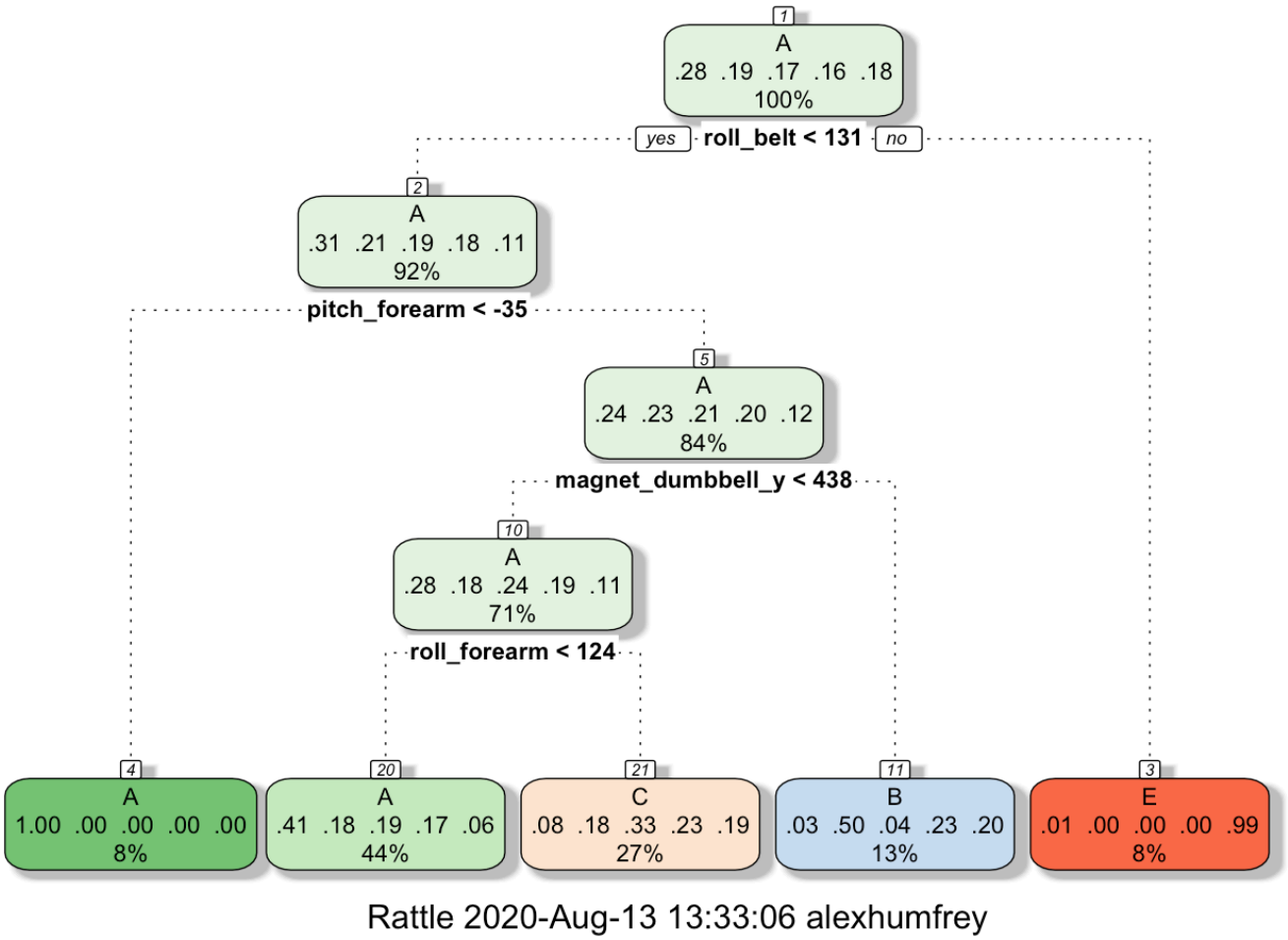
```
#predict on the validation set, and create confusions Matrix
predictRF <- predict(RFModel, newdata=valid_data)
cmrf <- confusionMatrix(predictRF, valid_data$classe)
cmrf$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1133    0    0    0    0
##          B    0  830    0    0    0
##          C    0    0  733    0    0
##          D    0    0    0  685    0
##          E    0    0    0    0  729
```

```
# classification tree  - model creation, model summary
set.seed(12345)
DTmodel <- train(classe ~ ., data=train_data, method="rpart")
DTmodel$finalModel
```

```
## n= 13737
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 130.5 12594 8697 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -34.55 1106    3 A (1 0.0027 0 0 0) *
##      5) pitch_forearm>=-34.55 11488 8694 A (0.24 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 437.5 9695 6962 A (0.28 0.18 0.24 0.19 0.11)
##         20) roll_forearm< 123.5 6037 3584 A (0.41 0.18 0.19 0.17 0.058) *
##         21) roll_forearm>=123.5 3658 2464 C (0.077 0.18 0.33 0.23 0.19) *
##       11) magnet_dumbbell_y>=437.5 1793  901 B (0.034 0.5 0.042 0.23 0.2) *
##    3) roll_belt>=130.5 1143    9 E (0.0079 0 0 0 0.99) *
```

```
fancyRpartPlot(DTmodel$finalModel)
```




```
#predict on validation data set
predictDT <- predict(DTmodel, newdata=valid_data)
cmdt <- confusionMatrix(predictDT, valid_data$classe)
cmdt$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1057  349  341  297   99
##          B   13  271   23  132  106
##          C   61  210  369  256  206
##          D    0    0    0    0    0
##          E    2    0    0    0  318
```

```
# generalised boosting model  - model creation, model summary
set.seed(1235)
controlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
GBMmodel  <- train(classe ~ ., data=train_data, method = "gbm",
                   trControl = controlGBM, verbose = FALSE)
GBMmodel$finalModel
```

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

```
#predict on validation data set
predictGBM <- predict(GBMmodel, newdata=valid_data)
cmgbm <- confusionMatrix(predictGBM, valid_data$classe)
cmgbm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1126   18    0    1    1
##          B    7  808   18    3    6
##          C    0    4  711   17    3
##          D    0    0    2  662    4
##          E    0    0    2    2  715
```

Model Selection and Prediction on Test Data Set

After assessing the estimated out of sample error for each model the Generalised Boosting Model was chosen as it has such a low error rate. The 100% accuracy of the Random Forest model is concerning and suggests there might be significant overfitting. The GBM was then used to predict the class from the test data set.

```
# Out of sample error for the three models
rf_error <- as.numeric(1- cmrf$overall["Accuracy"])
dt_error <- as.numeric(1- cmdt$overall["Accuracy"])
gbm_error <- as.numeric(1- cmgbm$overall["Accuracy"])

rf_error
```

```
## [1] 0
```

```
dt_error
```

```
## [1] 0.5097324
```

```
gbm_error
```

```
## [1] 0.02141119
```

```
# use chosen model on test data
predict_test <- predict(GBMmodel, newdata=test_data)
predict_test
```

```
##  [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
```