PMLpeergraded

Kawish Azad

24/10/2020

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.3
```

```r
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.3
```

```r
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.3
```

```r
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.3

## Loading required package: rpart
```

```r
library(rpart)
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.0.3

## Loaded gbm 2.1.8
```

```r
library(ggplot2)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3

## corrplot 0.84 loaded
```

Exloratory data analysis and data cleaning

```r
test_datalink <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
train_datalink  <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
test_data <- read.csv(url(test_datalink))
train_data <- read.csv(url(train_datalink))
```

now proceeding for the cleaning the input of the data

```r
training_dataset <- train_data[, colSums(is.na(train_data)) == 0]
testing_dataset <- test_data[, colSums(is.na(test_data)) == 0]
```

Splitting data into the ratio of 70 to 30 for train and test

```r
training_dataset <- training_dataset[, -c(1:7)]
testing_dataset <- testing_dataset[, -c(1:7)]
dim(training_dataset)
```

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

```r
set.seed(7717)
datatraining <- createDataPartition(train_data$classe, p = 0.7, list = FALSE)
training_dataset <- training_dataset[datatraining, ]
testing_dataset <- training_dataset[-datatraining, ]
dim(training_dataset)
```

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

```r
dim(testing_dataset)
```

```
## [1] 4122   86
```

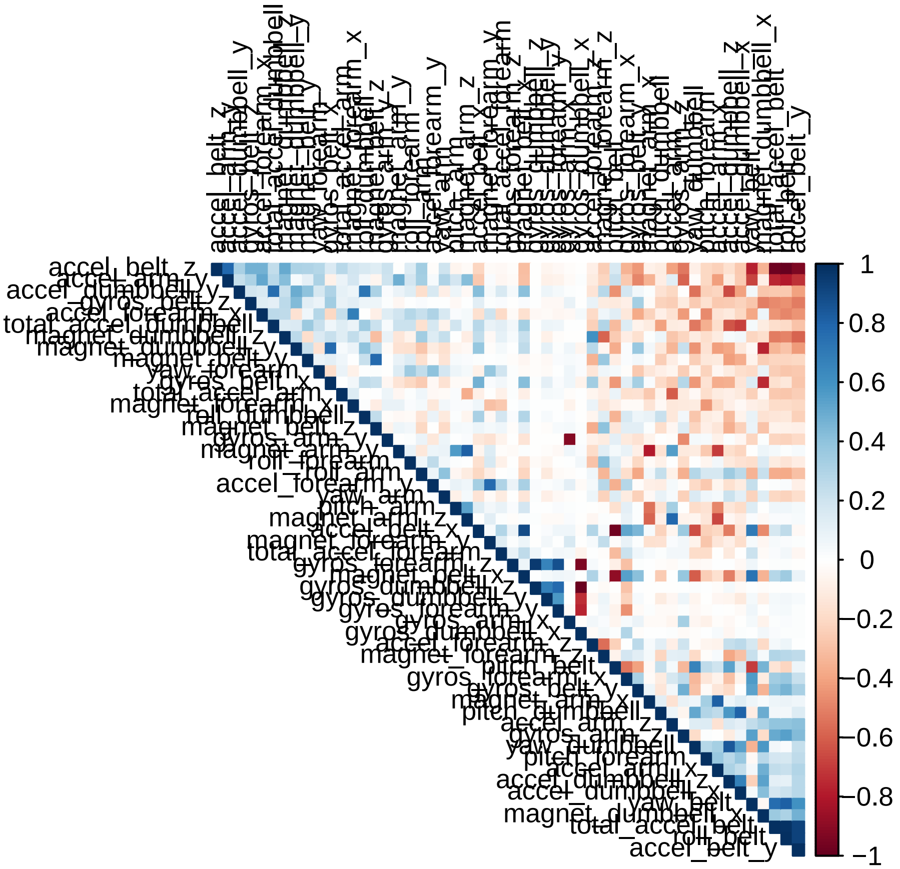
```r
noneZero <- nearZeroVar(training_dataset)
training_dataset <- training_dataset[, -noneZero]
testing_dataset <- testing_dataset[, -noneZero]
dim(training_dataset)
```

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

```r
dim(testing_dataset)
```

```
## [1] 4122   53
```

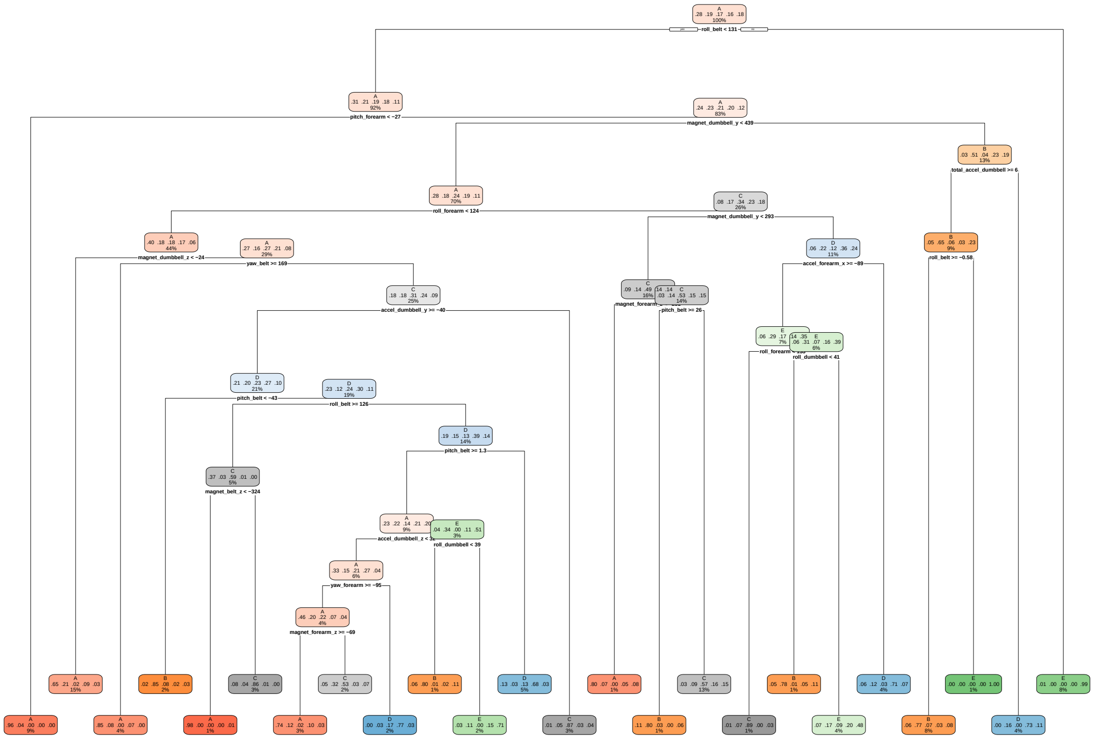
```r
plot_cor <- cor(training_dataset[, -53])
corrplot(plot_cor, order = "FPC", method = "color", type = "upper", tl.cex = 0.8, tl.col = rgb(0, 0, 0)
```



Now let's build the ML model for prediction

```
set.seed(1717)
X <- rpart(classe ~ ., data=training_dataset, method = "class")
rpart.plot(X)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



now we will be validate the model

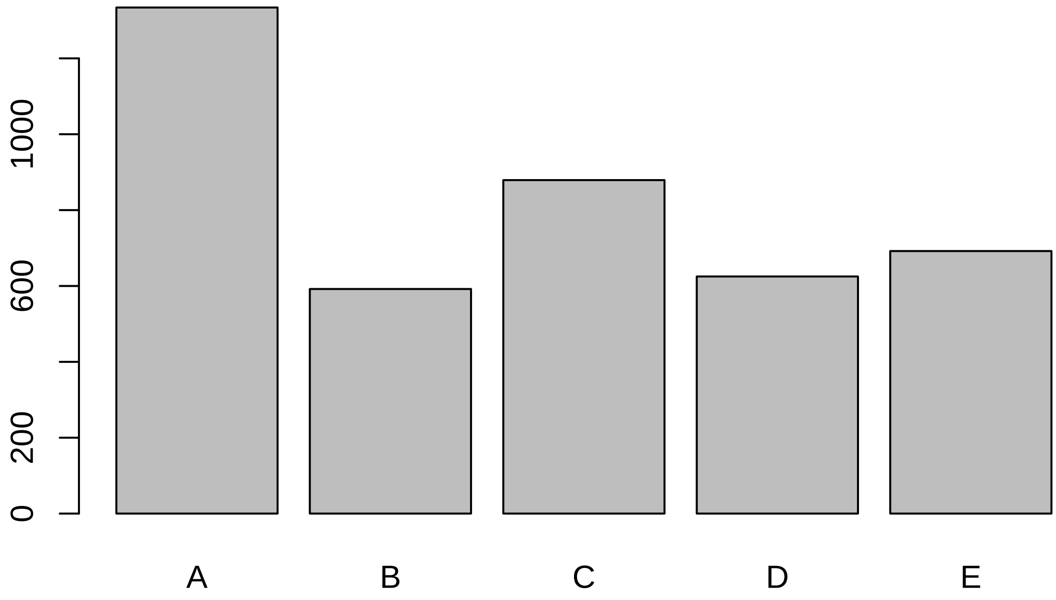
```
pred <- predict(X, testing_dataset, type = "class")
ab <- confusionMatrix(pred, as.factor(testing_dataset$classe))
ab
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1037  179   12   73   33
##          B   29  468   36   19   40
##          C   24   85  601   86   83
##          D   41   61   54  436   33
##          E   17   35   15   55  570
##
## Overall Statistics
##
##                Accuracy : 0.755
##                  95% CI : (0.7415, 0.768)
##     No Information Rate : 0.2785
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6892
##
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9033   0.5652   0.8370   0.6517   0.7510
## Specificity            0.9001   0.9624   0.9183   0.9453   0.9637
## Pos Pred Value         0.7774   0.7905   0.6837   0.6976   0.8237
## Neg Pred Value         0.9602   0.8980   0.9639   0.9334   0.9449
## Prevalence             0.2785   0.2009   0.1742   0.1623   0.1841
## Detection Rate         0.2516   0.1135   0.1458   0.1058   0.1383
## Detection Prevalence   0.3236   0.1436   0.2132   0.1516   0.1679
## Balanced Accuracy      0.9017   0.7638   0.8777   0.7985   0.8574
```

Now let's plot predictions

```r
plot(pred)
```



```r
set.seed(77777)
c_gbm <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
validation_gbm <- train(classe ~ .,data=training_dataset, method = "gbm", trControl = c_gbm, verbose = 
validation_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.
```