Prediction Assignment - Exercise Type

Vinicius Ranieri

8/16/2020

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. 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. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

The objective is to predict the exercise type A,B,C,D or E

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Data Processing

There are 2 dataset one will be used to train the model and the other to validate it.

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

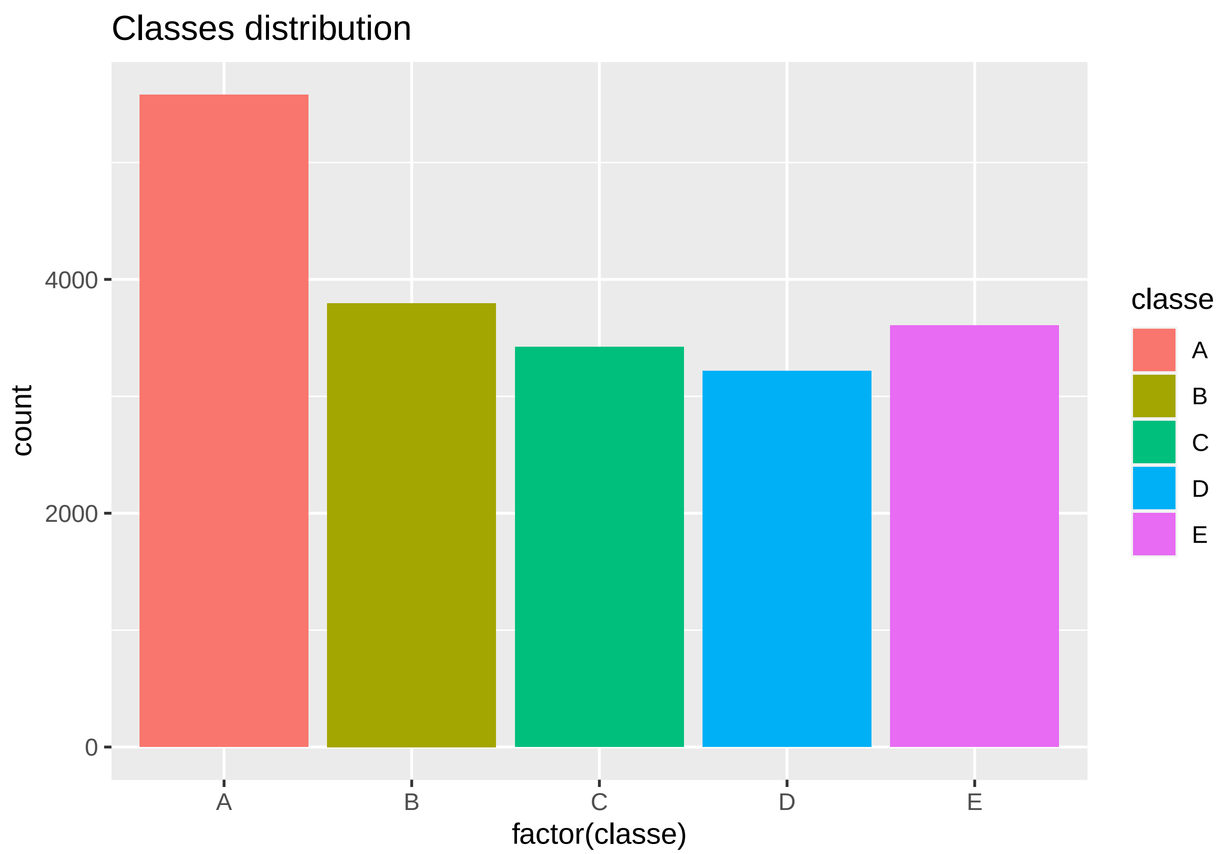
The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

```r
training <- read.csv("pml-training.csv")
testing <- read.csv("pml-testing.csv")

#Remove columns that have only NA values in the  testing dataset. Do it for both test and training data
training<-training[colSums(is.na(testing)) == 0]
testing<-testing[colSums(is.na(testing)) == 0]

clean_train<-training[-c(1,3,4,5,6,7)]
clean_test<-testing[-c(1,3,4,5,6,7)]

dim(training)
```

```
## [1] 19622    60
```

There are 19622 observations with 60 variables. Let´s take a look into the prediction variable distribution

```
library(ggplot2)
ggplot(training, aes(x=factor(classe),y=..count..,fill= classe))+
  geom_bar()+labs(title="Classes distribution")
```



It is possible to see that there is enough data to predict all the classe data as they are more than 10 times the number of variables, and also enough data to perform cross validation. To simplify the model variables, the ones that have almost no variance will be removed. This is done with the nearZeroVar function.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2
```

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

```
nvz <- nearZeroVar(training)

nvz
```

```
## [1] 6
```

All the columns have significant variance

Data Spliting and Cross Validation

The folds method will be used with 5 Folds for cross validation. The radomforest method will be used to train the model.

```
set.seed(1234)
folds<-createFolds(y=clean_train$classe,k=5)
myControl <- trainControl(method = "cv",
                          number = 5,
                          savePredictions = TRUE,
                          index = folds,
                          summaryFunction = defaultSummary) # just use accuracy to determine best model
fit1 <- train(classe ~ ., data = clean_train,method="rf",trControl = myControl)
fit1
```

```
## Random Forest
##
## 19622 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 3924, 3924, 3924, 3925, 3925
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9694349  0.9613155
##   29    0.9736521  0.9666592
##   57    0.9652177  0.9559862
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 29.
```

The model has a very good accuracy of 99.9% in our cross validation test. Let's check against the test data.

```
p <- predict(fit1, clean_train, type = "raw")
confusionMatrix(p, factor(clean_train$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5580    0    0    0    0
##          B    0 3797    0    0    0
##          C    0    0 3422    0    0
##          D    0    0    0 3216    0
##          E    0    0    0    0 3607
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
```

```
##     No Information Rate : 0.2844
##     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             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```r
quiz.p <- predict(fit1, clean_test, type = "raw")
quiz.p
```

```
##  [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
```