Prediction Assignment

guchiguchi

2018/5/19

libraries

```
library(tidyverse)
library(caret)
library(rattle)
```

```
## Warning: Failed to load RGtk2 dynamic library, attempting to install it.
```

check and preparing data

load csv file

```
pml_training <- read_csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",
   na = "0")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 185 parsing failures.
## row # A tibble: 5 x 5 col     row col           expected actual  file                    expected   <int> <chr>          <chr>
   <chr>   <chr>                     actual 1  2231 kurtosis_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudf…
file 2  2231 skewness_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudf… row 3  2255 kurtosis_roll_arm a
double #DIV/0! 'https://d396qusza40orc.cloudf… col 4  2255 skewness_roll_arm a double #DIV/0! 'https://d396qus
za40orc.cloudf… expected 5  2282 kurtosis_roll_arm a double #DIV/0! 'https://d396qusza40orc.cloudf…
## ... ................. ... ........................................................................ ........
........................................................................ ...... ....................................................................... ....
........................................................................ ... ....................................................................... ...
........................................................................ ........ .......................................................................
## See problems(...) for more details.
```

```
pml_testing <- read_csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
   na = "0")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

check dimention

```
dim(pml_training); dim(pml_testing)
```

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

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

```
pml_training %>% count(user_name, classe)
```

```
## # A tibble: 30 x 3
##    user_name classe     n
##    <chr>     <chr>  <int>
##  1 adelmo    A       1165
##  2 adelmo    B        776
##  3 adelmo    C        750
##  4 adelmo    D        515
##  5 adelmo    E        686
##  6 carlitos  A        834
##  7 carlitos  B        690
##  8 carlitos  C        493
##  9 carlitos  D        486
## 10 carlitos  E        609
## # ... with 20 more rows
```

check na's

```
count_na <- function(x){
  x %>% is.na() %>% sum()
}
data.frame(NA_count = sapply(pml_training,FUN = count_na), class = sapply(pml_training, class)) -> data_class
data_class %>% count(class)
```

```
## # A tibble: 3 x 2
##   class        n
##   <fct>    <int>
## 1 character   95
## 2 integer     32
## 3 numeric     33
```

convert class character to numeric(7:159)

```
ix <- 7:150
pml_training[ix] <- sapply(pml_training[ix], as.numeric)
```

Training data separate training and test.

We do not need "X1", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvtd_timestamp"

Delete them.

```
pml_training <-  pml_training[,6:160]
#pml_training$classe <- as.factor(pml_training$classe)
```

create training and testing data.

```
set.seed(20180520)
inTrain <- createDataPartition(y = pml_training$classe, p=0.7, list=FALSE)
training_training <- pml_training[inTrain,]
training_testing <- pml_training[-inTrain,]
```

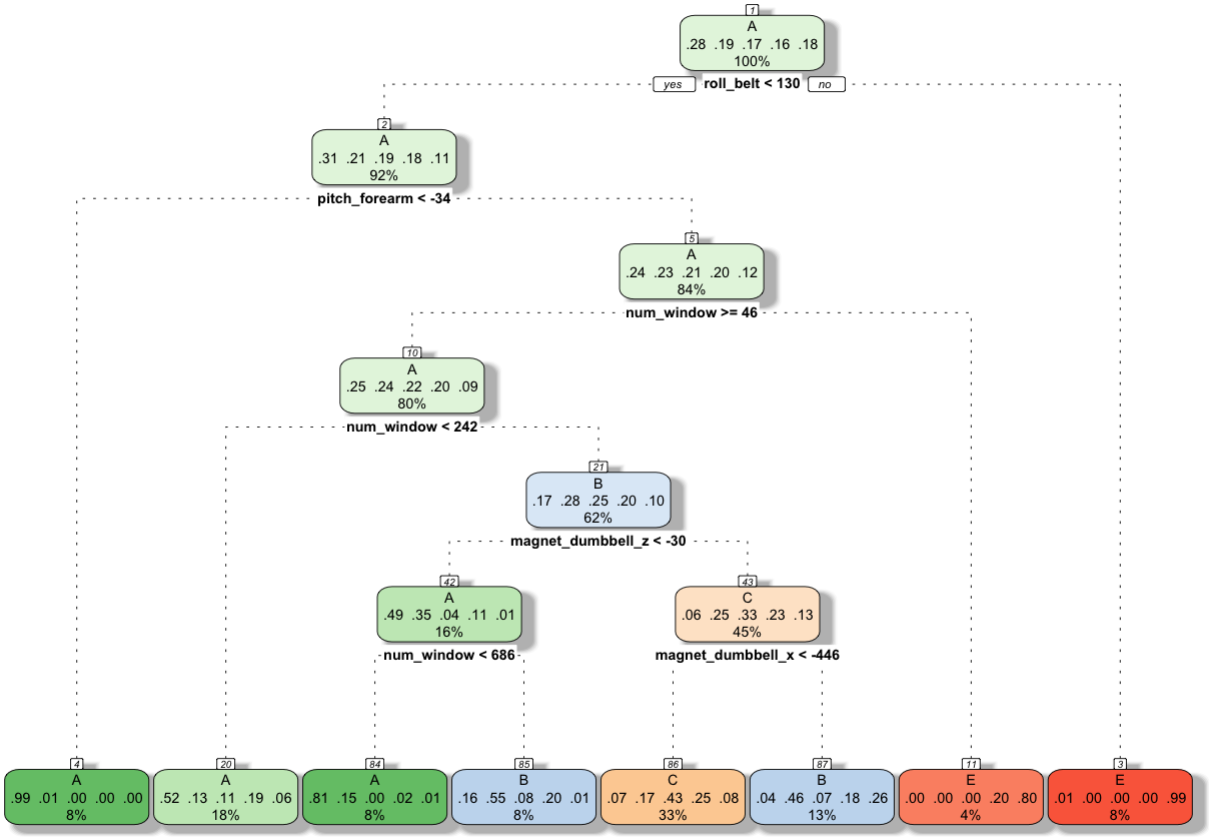
train by rapart.

```
training_training %>% mutate_all(funs(ifelse(is.na(.),0,.))) -> training_NaToZero
training_testing %>% mutate_all(funs(ifelse(is.na(.),0,.))) -> testing_NaToZero

mod_rapart <- train(data = training_NaToZero,
          classe ~.,
          method = "rpart")

print(mod_rapart$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 12598 8705 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -34.15 1112    6 A (0.99 0.0054 0 0 0) *
##      5) pitch_forearm>=-34.15 11486 8699 A (0.24 0.23 0.21 0.2 0.12)
##       10) num_window>=45.5 10970 8183 A (0.25 0.24 0.22 0.2 0.09)
##         20) num_window< 241.5 2508 1193 A (0.52 0.13 0.11 0.19 0.055) *
##         21) num_window>=241.5 8462 6126 B (0.17 0.28 0.25 0.2 0.1)
##           42) magnet_dumbbell_z< -29.5 2225 1145 A (0.49 0.35 0.043 0.11 0.0099)
##             84) num_window< 686.5 1109  206 A (0.81 0.15 0.0027 0.024 0.0072) *
##             85) num_window>=686.5 1116  504 B (0.16 0.55 0.082 0.2 0.013) *
##           43) magnet_dumbbell_z>=-29.5 6237 4202 C (0.063 0.25 0.33 0.23 0.13)
##             86) magnet_dumbbell_x< -445.5 4478 2560 C (0.073 0.17 0.43 0.25 0.084) *
##             87) magnet_dumbbell_x>=-445.5 1759  945 B (0.038 0.46 0.067 0.18 0.26) *
##       11) num_window< 45.5 516  104 E (0 0 0 0.2 0.8) *
##    3) roll_belt>=130.5 1139   13 E (0.011 0 0 0 0.99) *
```

file:///Users/aa575274/Documents/cousera/courses/08_PracticalMachineLearning/019predictingWithTrees/index.html#15
(file:///Users/aa575274/Documents/cousera/courses/08_PracticalMachineLearning/019predictingWithTrees/index.html#15)

```
fancyRpartPlot(mod_rapart$finalModel)
```



Rattle 2018- 6-03 16:37:13 aa575274

train by randomforest.

```
mod_rf <- train(data = training_NaToZero,
          classe ~.,
          trControl = trainControl(method="oob"), method = "rf")

# mod_gam <- train(data = a,
#           classe ~.,
#           method = "gam")

#xgboost
# mod_xgt <- train(data = a,
#           classe ~.,
#           method = "xgbTree")
#
# mod_xgl <- train(data = a,
#           classe ~.,
#           method = "xgbLinear")
```

check prediction

```
pred_rpar <- predict(mod_rapart, newdata = testing_NaToZero)
pred_rf <- predict(mod_rf, newdata = testing_NaToZero)
```

```
confusionMatrix(data = pred_rpar, testing_NaToZero$classe %>% as.factor())
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##         A 1378  208  118  214   55
##         B  139  632   81  259  200
##         C  156  299  827  449  161
##         D    0    0    0    0    0
##         E    1    0    0   42  666
##
## Overall Statistics
##
##               Accuracy : 0.5952
##                 95% CI : (0.5826, 0.6078)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.4833
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8232   0.5549   0.8060   0.0000   0.6155
## Specificity            0.8587   0.8569   0.7808   1.0000   0.9910
## Pos Pred Value         0.6984   0.4821   0.4371      NaN   0.9394
## Neg Pred Value         0.9243   0.8892   0.9502   0.8362   0.9196
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2342   0.1074   0.1405   0.0000   0.1132
## Detection Prevalence   0.3353   0.2228   0.3215   0.0000   0.1205
## Balanced Accuracy      0.8409   0.7059   0.7934   0.5000   0.8033
```

```
confusionMatrix(data = pred_rf, testing_NaToZero$classe %>% as.factor())
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##         A 1673    3    0    0    0
##         B    0 1135    2    0    0
##         C    0    0 1024    7    0
##         D    0    1    0  957    3
##         E    1    0    0    0 1079
##
## Overall Statistics
##
##               Accuracy : 0.9971
##                 95% CI : (0.9954, 0.9983)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.9963
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9965   0.9981   0.9927   0.9972
## Specificity            0.9993   0.9996   0.9986   0.9992   0.9998
## Pos Pred Value         0.9982   0.9982   0.9932   0.9958   0.9991
## Neg Pred Value         0.9998   0.9992   0.9996   0.9986   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1929   0.1740   0.1626   0.1833
## Detection Prevalence   0.2848   0.1932   0.1752   0.1633   0.1835
## Balanced Accuracy      0.9993   0.9980   0.9983   0.9960   0.9985
```

Random forest is good model

check testing data and answer the Course Project Prediction Quiz

prepare pml_testing data

```
ix <- 7:150
pml_testing[ix] <- sapply(pml_testing[ix], as.numeric)
pml_testing <-  pml_testing[,6:160]
pml_testing  %>% mutate_all(funs(ifelse(is.na(.),0,.))) -> pml_testing_NaToZero
```

predict Course Project Prediction Quiz

```
pred_rf_test <- predict(mod_rf, newdata = pml_testing_NaToZero)
pred_rf_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
```