Project

WendyD

5/28/2021

```
#load required package
library(caret)
```

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

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

```
library(randomForest)
```

```
## 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
```

Load data

```
downloadcsv <- function(url, nastrings) {
    temp <- tempfile()
    download.file(url, temp, method = "curl")
    data <- read.csv(temp, na.strings = nastrings)
    unlink(temp)
    return(data)
}

trainurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train <- downloadcsv(trainurl, c("", "NA", "#DIV/0!"))

testurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test <- downloadcsv(testurl, c("", "NA", "#DIV/0!"))
```

```
dim(train)
```

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

```
#proportion of each group in outcome variable
table(train$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

split train data into training and validation set 80 20

```r
set.seed(123456)
trainset <- createDataPartition(train$classe, p = 0.8, list = FALSE)
Training <- train[trainset, ]
Validation <- train[-trainset, ]
```

Remove zero variance variable, column with more than 40% missing value and " "

```r
# exclude near zero variance features
nzvcol <- nearZeroVar(Training)
Training <- Training[, -nzvcol]

# exclude columns with 40% ore more missing values exclude descriptive AKA has more than 60% valid valu
# columns like name etc
cntlength <- sapply(Training, function(x) {
    sum(!(is.na(x) | x == ""))
})    #sum of all row that has NO NA value or ""

# identify these column that has LESS than 60% valid values
nullcol <- names(cntlength[cntlength < 0.6 * length(Training$classe)])
descriptcol <- c("X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2",
    "cvtd_timestamp", "new_window", "num_window")   #columns to remove cause it add no value
excludecols <- c(descriptcol, nullcol)
Training <- Training[, !names(Training) %in% excludecols]
```

#Model Train

```r
Training$classe <- factor(Training$classe)
rfModel <- randomForest(classe ~ ., data = Training, importance = TRUE, ntrees = 10)
```

#Model Prediction

See how the model perform with validation set

```r
Validation$classe <- factor(Validation$classe)
pvalidation <- predict(rfModel, Validation)
print(confusionMatrix(pvalidation, Validation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    1    0    0    0
##          B    0  758    0    0    0
##          C    0    0  684    4    0
##          D    0    0    0  638    3
##          E    0    0    0    1  718
##
## Overall Statistics
##
##                Accuracy : 0.9977
##                  95% CI : (0.9956, 0.999)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9971
```

```
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9987   1.0000   0.9922   0.9958
## Specificity            0.9996   1.0000   0.9988   0.9991   0.9997
## Pos Pred Value         0.9991   1.0000   0.9942   0.9953   0.9986
## Neg Pred Value         1.0000   0.9997   1.0000   0.9985   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1932   0.1744   0.1626   0.1830
## Detection Prevalence   0.2847   0.1932   0.1754   0.1634   0.1833
## Balanced Accuracy      0.9998   0.9993   0.9994   0.9957   0.9978
```

The accuracy for validation set is 99.7% so our model is doing pretty goid

Predict Test set

```
ptest <- predict(rfModel, test)
ptest
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

Export our prediction to answer key

```
answers <- as.vector(ptest)

pml_write_files = function(x) {
    n = length(x)
    for (i in 1:n) {
        filename = paste0("problem_id_", i, ".txt")
        write.table(x[i], file = filename, quote = FALSE, row.names = FALSE,
            col.names = FALSE)
    }
}

answer_key <- pml_write_files(answers)
answer_key
```

```
## NULL
```