Practical machine learning project

Fuyu

July 21, 2020

OVerview

In this project, we first download the data and split the train dataset to two separate datasets:training and testing. Because the dimensions of the data is so huge, and there is a lot of missing values. We need to drop these missin values and reduce the dimension of predictors. Then we fit a classification model and test it on the test dateset. Finally we apply the model to the new data and make predictions.

```
path<-getwd()
url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(url, file.path(path, "train.csv"))
url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(url, file.path(path, "test.csv"))
train<-read.csv("train.csv")
test<-read.csv("test.csv")
library(caret)
```

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

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

```
library(AppliedPredictiveModeling)
library(dplyr)
```

```
##
## 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
```

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

```
## Loading required package: rpart
```

```
library(rpart)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## XXXX 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
```

```
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
names(train)[names(train)=="X"]="class"
names(test)[names(test)=="X"]="class"
train$class<-as.factor(train$class)
test$class<-as.factor(test$class)
```

Exclude missing value

```
drop_NA<-function(df){
        index<-NULL
        for( i in 1:dim(df)[2] ){
                if(is.na(df[1,i])==T){index<-c(index,i)}
        }
        df<-df[,-c(1,2,index)]
        }
train1<-drop_NA(train)
test1<-drop_NA(test)
```

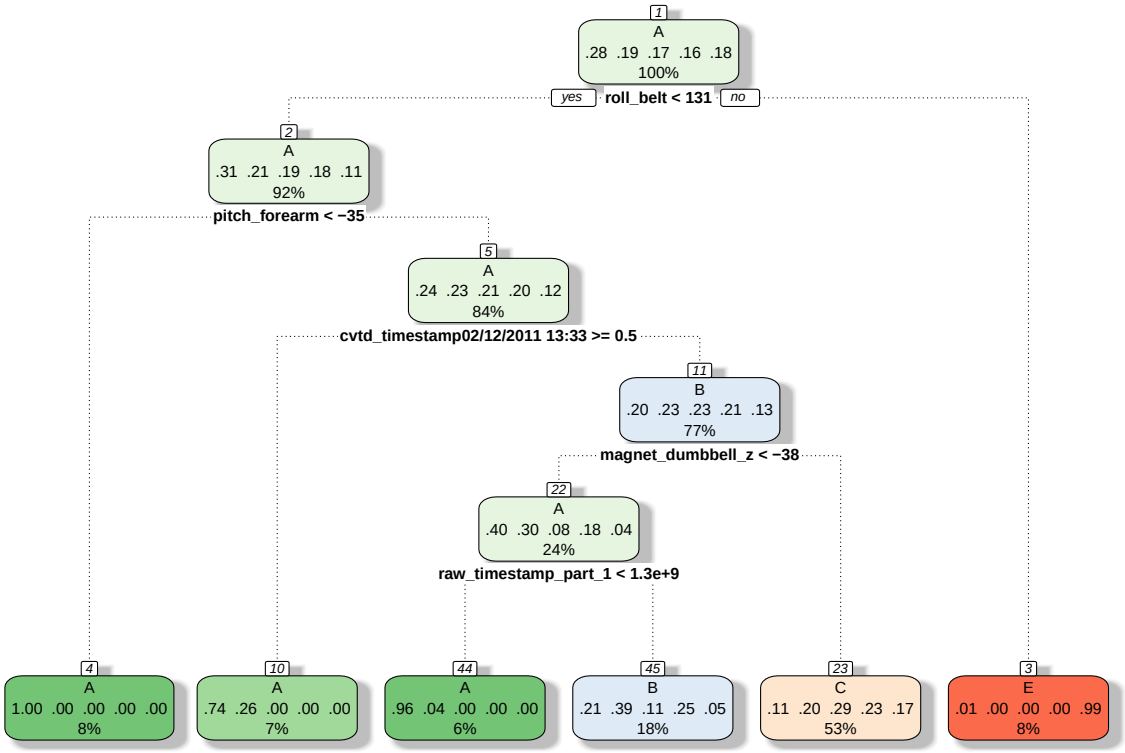
Reduce the dimension of predictors

```
intrain<-createDataPartition(y=train1$classe,p=0.7,list=F)
train1_train<-train1[intrain,]
train1_test<-train1[-intrain,]
NZV <- nearZeroVar(train1_train)
train1_train <- train1_train[, -NZV]
train1_test <- train1_test[, -NZV]
dim(train1_train)
```

```
## [1] 13737    57
```

Build a classification trees

```
set.seed(1)
mod<-train(classe~.,data=train1_train,method="rpart")
fancyRpartPlot(mod$finalModel)
```



Crossvalidate the model

```
pred<-predict(mod,train1_test)
tb<-table(pred,train1_test$classe)
confusionMatrix(tb)
```

```
## Confusion Matrix and Statistics
##
##
## pred    A    B    C    D    E
##    A 1059  114    0    0    0
##    B  212  422  114  249   77
##    C  400  603  912  715  521
##    D    0    0    0    0    0
##    E    3    0    0    0  484
##
## Overall Statistics
##
##                Accuracy : 0.4889
##                  95% CI : (0.476, 0.5017)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3606
##
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.6326  0.37050   0.8889   0.0000  0.44732
## Specificity            0.9729  0.86262   0.5392   1.0000  0.99938
## Pos Pred Value         0.9028  0.39292   0.2894      NaN  0.99384
## Neg Pred Value         0.8695  0.85097   0.9583   0.8362  0.88922
## Prevalence             0.2845  0.19354   0.1743   0.1638  0.18386
## Detection Rate         0.1799  0.07171   0.1550   0.0000  0.08224
## Detection Prevalence   0.1993  0.18250   0.5354   0.0000  0.08275
## Balanced Accuracy      0.8028  0.61656   0.7140   0.5000  0.72335
```

Therefore, here the **accuary** is **0.6503** and the **out of sample error** is **0.35**

Applying the model to the test data

```
predict(mod,test1)
```

```
##  [1] B C C A C C C C A A C C C A C C C B B C
## Levels: A B C D E
```