PML Project (Weight Lifting Exercise)

Wai Yan Moe Lwin

12/22/2022

Introduction

One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, our goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and build a model to predict how well they do the exercise.

We can get the training data from here and testing data from here.
All the data are from sourced from here

Transforming Data

First, we will load the data and required packages.

```
library(caret)
```

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

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

```
training <- read.csv('pml-training.csv')
testing <- read.csv('pml-testing.csv')
dim(training); dim(testing)
```

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

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

Now, when we look at the data, there are variables with lots of missing values. So, we decided to exclude them from our model. According to our purpose to predict whether a weight lifting rep is good or not from sensors, we will also exclude variables which are not from sensors.

```
not_na_col <- colSums(is.na(testing))==0
training <- training[not_na_col]
testing <- testing[not_na_col]

training <- training[8:60]
testing <- testing[8:60]

InTrain <- createDataPartition(training$classe, p=0.7, list=F)
Train <- training[InTrain,]
Test <- training[-InTrain,] #This is for out of sample validation.
```

Now, we fit our model with random forest. It took around 2 hours for me to fit it so I will save it and load it for future use.

```
if (file.exists('model.rds')){
  fit <- readRDS('model.rds')
}else{
  fit <- train(classe~., 'rf', data=Train)
  saveRDS(fit, 'model.rds')
}
```

Then, we will look into accuracies.

```
train_pred <- predict(fit, Train)
train_accuracy <- mean(train_pred==Train$classe)
train_accuracy
```

```
## [1] 0.9980345
```

```
table(train_pred, Train$classe)
```

```
##
## train_pred    A    B    C    D    E
##          A 3904    7    0    0    0
##          B    2 2647    3    0    1
##          C    0    4 2390    6    1
##          D    0    0    3 2246    0
##          E    0    0    0    0 2523
```

```
test_pred <- predict(fit, Test)
test_accuracy <- mean(test_pred==Test$classe)
test_error <- 1 - test_accuracy
test_accuracy
```

```
## [1] 0.9984707
```

```
table(test_pred, Test$classe)
```

```
##
## test_pred    A    B    C    D    E
##         A 1673    3    0    0    0
##         B    1 1136    1    0    0
##         C    0    0 1024    2    0
##         D    0    0    0  961    0
##         E    0    0    1    1 1082
```

So, we can see that in-sample accuracy is 0.9980345 out-of-sample accuracy is 0.9984707. Therefore, estimated out-of-sample error is 0.0015293 and we conclude our project here.