practicalML_proj_sd

2022-08-24

Overview

The Goal of the project is to predict "classse" variable which is a character variable with 5 levels ("A" "B" "C" "D" "E"). In this dataset, "Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E)" (Velasso et al. 2013).

How I used cross validation:

I conducted cross validation by splitting the "training set" into cvTraining and cvTesting sets. Then, I built a model vased on the cvTraining set, and tested on the cvTesting set. I repeated and averaged the estimated errors.

The purpose of cross-validation is to pick variables, the prediction models, and the parameters to include in our model.

What I think the expected out of sample error is:

It is the error rate one gets on a new data set using a model. It is sometimes referred to as "generalization error". I expect this generalization error to be very small since our training set is relatively large with 19622 observations. A model that is build on this large data set should be reliable enough to produce a small out of sample error rate on a new data set.

describe how I built the model

I downloaded and imported data into R First, I downloaded the training and testing sets from the following links: - training set: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv - test set: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

After downloading these data sets, I uploaded the files into R:

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

Load libraries used in this analysis

```
library(caret); library(randomForest); library(rpart.plot); library(rpart); library(ggplot2)
```

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

```
## Loading required package: lattice

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## Loading required package: rpart
```

Cleaning the data:

Cleaning a) Remove variables from the training set that have >50% of data as NAs. Repeat this for the testing set.

```
dim(training) # pre-cleaning dimensions
```

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

```
training <- training[ , which( colMeans(!is.na(training))>0.5 )]
dim(training) # post-cleaning dimensions
```

```
## [1] 19622    93
```

```
# now make sure the training set and testing sets have the same number of columns
dim(testing) # pre-cleaning dimensions
```

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

```
keep <- c(colnames(training)) # column names in training after cleaning
testing = testing[,(names(testing) %in% keep)]
dim(testing) # post-cleaning dimensions. "classe" variable is missing in testing set
```

```
## [1] 20 92
```

Cleaning b) remove variables from the training set that have very small variances. Repeat this for the testing set.

```
badCols <- nearZeroVar(training)
badCols # index of bad columns to be removed
```

```
##  [1]  6 12 13 14 15 16 17 18 19 20 43 44 45 46 47 48 52 53 54 55 56 57 58 59 60
## [26] 74 75 76 77 78 79 80 81 82
```

```
training <- training[, -badCols]
testing <- testing[, -badCols]

names(training)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "num_window"
##  [7] "roll_belt"            "pitch_belt"           "yaw_belt"
## [10] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
## [13] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [16] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [19] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [22] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [25] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [28] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [31] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [34] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [37] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [40] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [43] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [46] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [49] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [52] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [55] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [58] "magnet_forearm_z"     "classe"
```

```
dim(training) # dimensions of cleaned training set
```

```
## [1] 19622    59
```

```
dim(testing) # dimensions of cleaned testing set
```

```
## [1] 20 58
```

Partition the training set

Partition the "training set" into 2 sets for cross-validation purposes. The original training set was split into training (60 % of training set) and validation (40 % of training set).

```
inTrain <- createDataPartition(y=training$classe, p=0.4, list=FALSE)
training <- training[inTrain, ];
validation <- training[-inTrain, ]
dim(training); dim(validation); dim(testing)
```

```
## [1] 7850   59
```

```
## [1] 4728   59
```

```
## [1] 20 58
```

Attemptred a feature plot now to see how the variables are related to each other, but the there were many warnings about NAs withing the variables.The following is the code used:

featurePlot(x = training, y = training$classe)

Building prediction models

I decided to built a Random Forest to predict the "classe" variable.

```
mod <- train(classe ~., method = "rf", data = training)
mod
```

```
## Random Forest
##
## 7850 samples
##   58 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 7850, 7850, 7850, 7850, 7850, 7850, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9895788  0.9868191
##   41    0.9997506  0.9996847
##   80    0.9996261  0.9995273
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 41.
```

Predicting 20 different test cases using the testing set

```
#pred <- predict(mod, newdata = testing, class = "class")
#cfmatrix <- confusionMatrix(pred, as.factor(testing$classe))
#cfmatrix
```

Citations

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.