Peer-graded Assignment: Machine Learning Project

Guilherme S Lopes

6/11/2020

Overview

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with.

You should create a report describing:

1. how you built your model,
2. how you used cross validation,
3. what you think the expected out of sample error is, and
4. why you made the choices you did.
5. You will also use your prediction model to predict 20 different test cases.

(1) How I built my model

First, I initiate the relevant libraries, open the training dataset, and check its rows and columns.

```
set.seed(999)
suppressMessages(library(caret))
```

```
## Warning: package 'caret' was built under R version 3.6.3
```

```
suppressMessages(library(rpart))
suppressMessages(library(randomForest))
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
dim(training)
```

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

The dataset has 19622 rows and 160 variables. The outcome variable is “classe”.

```
table(training$classe)
```

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

Machine learning algorithms can be chosen based on the characteristics of the outcome (among many other things). Generalized linear modelling ("glm") is not very suitable for our case because the outcome "classe" has 5 categories (and glm is more adequate for continuous or binary).

Therefore, I wll create two models using random forests and decision trees (algorithms that can easily incorporate outcomes with multiple categories). The model with best fit will be tested on the testing dataset.

Before we train the models, there are many variables that are irrelevant or that contain missing in this dataset. I cleaned the training dataset by assigning NAs where appropriate and by removing irrelevant variables plus all variables with missing values.

```
training <- training[, -c(1:7)]
training[training == "#DIV/0!"] <- NA

cols <- as.data.frame(colnames(training))
for(i in 1:length(training))
  {cols$uniq[i] <- sum(as.data.frame(table(unique(training[i])))$Freq)}

training <- training[, -c(7, 10, 19, 82, 85, 94, 120, 123, 132)]
training <- training[, colSums(is.na(training)) == 0]
rm(cols, i)
```

The dataset is now ready for machine learning analysis.

NOTE: There are more robust ways to clean the data beyond the purposes of this assignment.

(2) How I used cross validation

This dataset has a large sample size that can afford splitting. For this reason, I will perform a cross validation by splitting the training dataset in two:

- training_1 (60% of training dataset)
- training_2 (40% of training dataset)

```
inTrain = createDataPartition(training$classe, p = 6/10)[[1]]
training_1 = training[ inTrain,]
training_2 = training[-inTrain,]
```

I will predict training_2 based on the modelling of training_1.

Model 1 uses random forests:

```
fit_1 <- randomForest(classe ~ ., data = training_1, method = "class")
predict_1 <- predict(fit_1, training_2, type = "class")
```

Model 2 uses decision trees:

```
fit_2 <- rpart(classe ~ ., data = training_1, method = "class")
predict_2 <- predict(fit_2, training_2, type = "class")
```

(3) What I think the expected out of sample error is

Accuracy is a percentage value, and corresponds to the percentage of predicted values that are identical to the true values among all true values.

The expected error is then the remaining percentage of predicted values that failed to predict correctly (1 - Accuracy).

That is, 1 - Accuracy is the error we expect to see whe using our model.

(4) Why I made the choices I did

My choice of decision trees and random forests (among dozens of algorithms) was due to the nature of the outcome (categorical variable with > 2 classifiers) plus the fact that those algorithms are very common in the data science community.

I decided to compare the models in terms of accuracy [95%CI] because accuracy is a reliable and easily interpretable model fit parameter.

Random forests:

```
cat(
"Accuracy [95%CI] = ",
round(confusionMatrix(predict_1, training_2$classe)$overall[1],3), " [",
round(confusionMatrix(predict_1, training_2$classe)$overall[3],3), "-",
round(confusionMatrix(predict_1, training_2$classe)$overall[4],3), "]", sep =
"")
```

```
## Accuracy [95%CI] = 0.992 [0.99-0.994]
```

Decision trees:

```
cat(
"Accuracy [95%CI] = ",
round(confusionMatrix(predict_2, training_2$classe)$overall[1],3), " [",
round(confusionMatrix(predict_2, training_2$classe)$overall[3],3), "-",
round(confusionMatrix(predict_2, training_2$classe)$overall[4],3), "]", sep =
"")
```

```
## Accuracy [95%CI] = 0.741 [0.731-0.75]
```

Clearly, the random forests model has a better fit (i.e. it has higher accuracy).

(5) I also used my prediction model to predict 20 different test cases.

First, I open the testing dataset.

Then, I apply our random forest model to the testing dataset.

```
predict_test <- predict(fit_1, testing, type = "class")
predict_test
```

```
##  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
```

The values we see above are EXPECTED categories based on our model.

Please note:

The data curation and machine learning practice above were simplistic, and in a real setting the data science team would delve into the data with much more rigor. Every variable would be scrutinized in terms of quality and relevance. Different algorithms (and different models within each algorithm) would be compared and discussed among the team members before deciding on the final model.

I really appreciate the time and effort you invested in providing feedback on this assignment!

Thank you.