Peer assessment ML

Rene int Veld

Thursday, December 8, 2016

Introduction project

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 in an experiment, and then predict the manner in which they did their exercises. This is the “classe” variable in the training set. We will use any of the other variables to predict with. Underneath we will be describing how we built our model, how we used cross validation, and what is the expected out of sample error is, and why we made the choices we have made. Furthermore, we will use your prediction model to predict 20 different test cases.

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

The HAR (Human Activity Recognition) Dataset contains 5 classes (sitting-down, standing-up, standing, walking, and sitting) collected on 8 hours of activities of 4 healthy subjects. We also established a baseline performance index. Read more: http://groupware.les.inf.puc-rio.br/har#ixzz4RhWds5wi (http://groupware.les.inf.puc-rio.br/har#ixzz4RhWds5wi)

Loading and preprocessing the data

Data: The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv) The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). We would like to thank the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), as they have been very generous in allowing their data to be used for this kind of assignment.

We have downloaded the above mentioned 2 files and stored it in local environment for this peer assessment:

```
setwd("~/R/working directory course/predicting activity")

url.train <-
    "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
url.test <-
    "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

download.file(url.train, "train.csv")
download.file(url.test, "test.csv")
```

Now we have to read the csv files into R-files for further processing.

```
act.train <- read.csv("train.csv")
act.test <- read.csv("test.csv")
```

We know have a training database of 19662 observations and 160 variables and a testing database of only 20 observations. Note that the observations were made between 28 November and 5 December of 2011, for as well the training as the testing database: which means that we do not make a prediction of future observation (ie. now time series involved). We can get an overview of the 6 participants and the number of observations in both sets:

```
table(act.train$user_name)
```

```
##
##   adelmo carlitos  charles   eurico   jeremy    pedro
##     3892     3112     3536     3070     3402     2610
```

```
table(act.test$user_name)
```

```
##
##   adelmo carlitos  charles   eurico   jeremy    pedro
##        1        3        1        4        8        3
```

We noted that all field names are the same, except for the last field. In the training set the last field is called Classe, this is the score describing how well the exercise was performed. In the test set this field is replaced by problem-id, this is the number of the observation of which a prediction has to be made of the score. This can be verified as follows:

```
table(act.train$classe)
```

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

```
table(act.test$problem_id)
```

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

Now we have to predict the scores of the test set. We noted that many of the 160 columns do not have a valid result (eg. blank or NA or #DIV/0), especially in the test set, so they can be omitted from both sets. These are fields that represent statistical measures: -amplitude -avg -kurtosis -max -min -skewness -stddev -var All these fields starting with above terms have been deleted in the train and test set to create new sets on which we will make our prediction.

```
x <- names(act.train)
fields.to.select <- subset(x, ! substr(x[],1,3) %in% c("amp","avg","kur","ma
x","min","ske","std","var"))
train <- subset(act.train, select = fields.to.select)
```

This way we only analyse the 60 relevant fields for our prediction.

Some variables are factors, we will transform them: - new_window is a almost zero variable, we will remove this - cvtd_timestamp is based on the raw_timestramps, we can remove this - user_name has to be converted to 6 indicator-values - and the classe A-E, which we assume to be a Brazilian grade ranging from A to E, can be converted to a numeric between 0 and 10: Grade Scale Grade Description A 9 - 10 Excelente (Exellent) –> 10 B 7 - 9 Bom (Good) –> 8 C 5 - 7 Aceptable (Average) –> 6
D 3 - 5 Suficiente (Sufficient) –> 4 E 0 - 3 Defieciente (Fail)–> 2

```
train$new_window <- NULL
train$cvtd_timestamp <- NULL
train$adelmo <- ifelse(train$user_name=="adelmo",1,0)
train$carlitos <- ifelse(train$user_name=="carlitos",1,0)
train$charles <- ifelse(train$user_name=="charles",1,0)
train$eurico <- ifelse(train$user_name=="eurico",1,0)
train$jeremy <- ifelse(train$user_name=="jeremy",1,0)
# train$pedro <- ifelse(train$user_name=="pedro",1,0) would cause dependency
train$user_name <- NULL
levels(train$classe) <- c(10, 8, 6, 4, 2)
train$grade <- as.numeric(as.character(train$classe))
train$classe<- NULL
```

Choosing a prediction method

We can now choose a prediction method to predict the scores in the test set. For this we need several libraries, eg. caret, etc. etc.

```
library(caret)
```

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

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

```
library(AppliedPredictiveModeling)
library(CORElearn)
library(e1071)
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Versión 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Escriba 'rattle()' para agitar, sacudir y  rotar sus datos.
```

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

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

```
library(ElemStatLearn)
library(pgmm)
library(rpart)
library(gbm)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.1
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(forecast)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: timeDate
```

```
##
## Attaching package: 'timeDate'
```

```
## The following objects are masked from 'package:e1071':
##
##     kurtosis, skewness
```

```
## This is forecast 7.3
```

```
library(ggplot2)
library(ISLR)
```

First we will do some Cross Validation. For this we will split the train-set in a training and a testing set (the latter is not the test set on which we will make our final predictions). We will build the model on the training set and then evaluate the model based on the testing set. The predicted values are continous - for instance 8.0326 - and we need to transform these values back into a integer representing a grade, ie. 2,4,6,8,10.

```
inTrain <- createDataPartition(y=train$grade,
                               p=0.75, list=FALSE)

training <- train[inTrain,]
testing <- train[-inTrain,]

set.seed(32343)

# use Generalized Linear Model
modelFit <- train(grade~ .,method="glm",data=training)
modelFit
```

```
## Generalized Linear Model
##
## 14718 samples
##    61 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 14718, 14718, 14718, 14718, 14718, 14718, ...
## Resampling results:
##
##   RMSE       Rsquared
##   0.1769934  0.9964136
##
##
```

```
pred.raw <- predict(modelFit, newdata=testing)
pred <- round(pred.raw/2)*2

# the transformation from pred.raw to pred is to turn the continuous estimat
ion in a integer representing a grade, ie. 2,4,6,8,10

table(pred)
```

```
## pred
##    2    4    6    8   10
##  920  785  855  980 1364
```

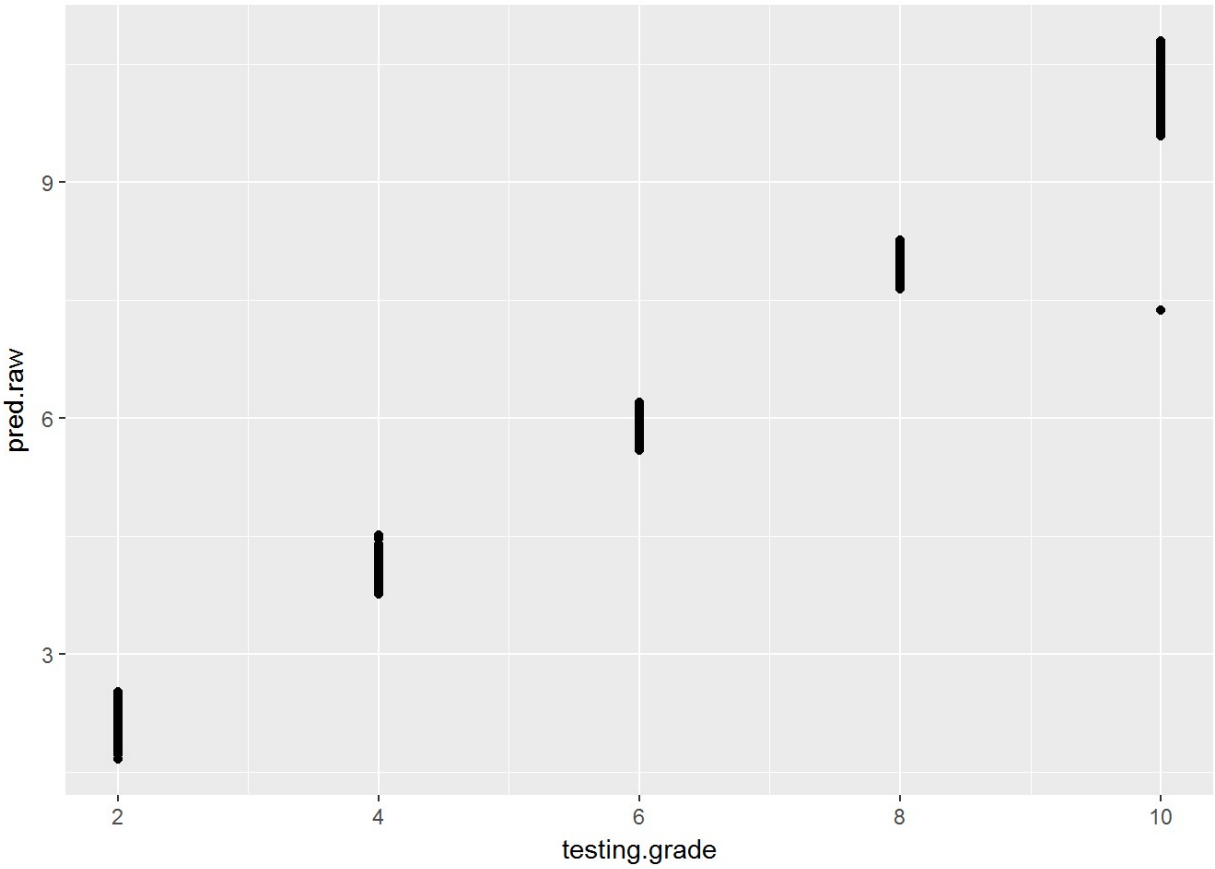
```
table(testing$grade)
```

```
##
##    2    4    6    8   10
##  920  785  855  979 1365
```

The prediction is almost 100% if you look at the distribution. We can plot the individual raw predictions versus the real grades in a scatter diagram. From the diagram you can see that there are only two outliers in the data, where the raw prediction is between 6 and 7 and this is estimated as grade 6 which in reality lead to a grade 8.

Furthermore we note that there is a strange discontinuity in the raw predictions resembling the real grads, which could point into the direction of overfitting. Normally you would expect a continuous result between 0 and 10. However, another possibility is that the exercises are extended until a clear score has been established (maybe a bit far-fetched).

```
graph.data <- data.frame(testing$grade,pred.raw)
qplot(testing.grade, pred.raw, data=graph.data)
```



Final prediction

We can do now the prediction of the final test set with the glm-method

```
# use Generalized Linear Model
modelFit <- train(grade ~ .,method="glm", data=train)
modelFit
```

```
## Generalized Linear Model
##
## 19622 samples
##    61 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 19622, 19622, 19622, 19622, 19622, 19622, ...
## Resampling results:
##
##   RMSE       Rsquared
##   0.1784596  0.9963419
##
##
```

```
# use Generalized Linear Model with preprocessing
# modelFit <- train(grade ~ .,method="glm", preProcess="pca", data=train)
# modelFit

# use Random Forest Model
# modelFit <- train(grade ~ .,method="rf",data=train) does no work because
5 or fewer unique values (grades)
```

We can see that the glm method with pre-processing is less accurate than without, so we will prefer the glm method without pre-processing.

We also tried the random forest method, but this method does not work with only 5 possible outcomes (the 5 possible grades). We tried some other models like lasso and lm, but none of them had a lowere RMSE than the glm-method.

Conclusion

We can conclude that the prediction made might be too good, ie. could be overfitting, so that the results on the final test set could be a bit disappointing.