PracticalML Course Project

Bintu G. vasudevan

Sunday, September 14, 2014

Executive Summary

The goal of the project is to use machine learning algorithm and predict the class in weight lifting exercises dataset using the best explanatory variables.

Weight Lifting Exercises Dataset

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.

Six young health participants were asked to perform barbell (lifts correctly and incorrectly in 5 different ways) set of 10 repetitions, 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). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate.

In this project, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and predict the specified execution either of Class (A,B,C,D,E).

More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Dataset reference: "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."

The har data consist of 19622 data points collected on 160 predictors. The goal is to predict the 5 classes (A,B,C,D,E).

Filtering the Data

After pre processing the data for removal of the "#DIV/0!" and "NA", finally got 19622 data points with 60 predictors, and further selecting data which are related the belt, forearm, arm, and dumbell (removing the timestamp and date). Final there is set of 52 predictors.

Model Building Approach

We have given a training data for this project we will use this training data and first, split the data into two groups: a training set and a test set. The given Testing data is used for Validation.

After applying the the filter to data we split the data into two groups, the createDataPartition function.

The dim(training) = 14718 of 53 predictors and dim(testing) = 4904 53 predictors. (predictors incluing the classe)

The partial least squares discriminant analysis (PLS-DA) method has been used for predicting the results. Here we have used “classe” as the response varial and all the rest 52 variables as predictors.

A trainControl function is used with the option method controls the type of resampling and used “repeatedcv”, this is used to specify repeated K{fold cross_validation (and the argument repeats controls the number of repetitions). K is controlled by the number argument and defaults to 10. All the data have been normalized and scaled.

ctrl <- trainControl(method = “repeatedcv”, repeats = 3)

plsFit <- train(classe ~ ., data = training, method = “pls”, tuneLength = 15, trControl = ctrl, preProc = c(“center”, “scale”))

```
load("my_model1.rda")
testing = read.csv("testing.csv")

library(caret)
```

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

```
data(plsFit)
```

```
## Warning: data set 'plsFit' not found
```

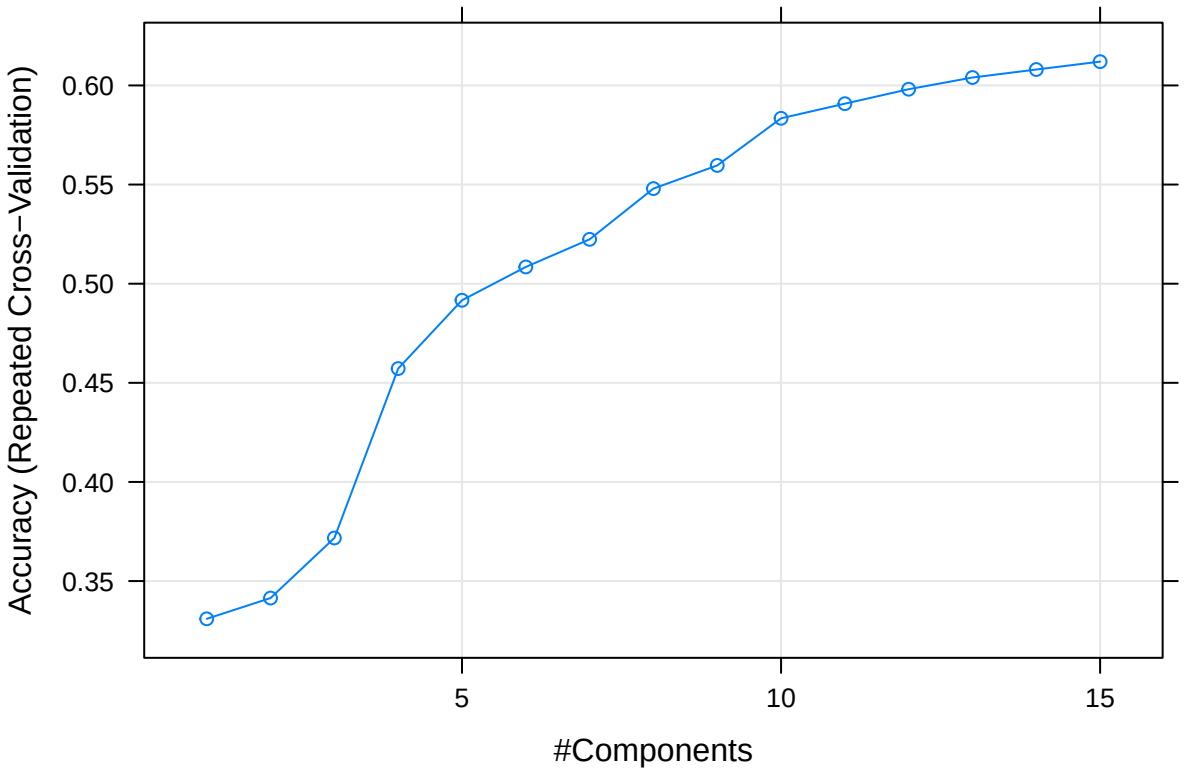
```
confusionMatrix(testing$classe,predict(plsFit,newdata=testing))
```

```
## Loading required package: pls
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:caret':
##
##     R2
##
## The following object is masked from 'package:stats':
##
##     loadings
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1183   25   54  106   27
##          B  255  425  139   51   79
##          C  226   57  427  108   37
##          D  113   48   58  509   76
##          E   81  132  100  131  457
##
## Overall Statistics
##
##                Accuracy : 0.612
##                  95% CI : (0.598, 0.626)
```

```
##     No Information Rate : 0.379
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.504
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.637   0.6186   0.5488    0.562   0.6760
## Specificity             0.930   0.8757   0.8963    0.926   0.8950
## Pos Pred Value          0.848   0.4478   0.4994    0.633   0.5072
## Neg Pred Value          0.808   0.9338   0.9133    0.903   0.9453
## Prevalence              0.379   0.1401   0.1586    0.185   0.1378
## Detection Rate          0.241   0.0867   0.0871    0.104   0.0932
## Detection Prevalence    0.284   0.1935   0.1743    0.164   0.1837
## Balanced Accuracy       0.784   0.7472   0.7226    0.744   0.7855
```

```
plot(plsFit)
```



The relationship between the resampled performance values and the number of PLS components. The command plot(plsFit) produced the results as shown in Figure 1. For ploting purpose the data was load, apply filter and then build the model and the model file were saved. This save model file is loaded for fast executing of the HTML markdown file.

Predicting on the test dataset

We load the given test data for the course project assignment and so the same filtering and then load the fitted model to run on the rtest data. Below is the R code which si use to run on the test dataset

```
harTestdat = read.csv("pml-testing.csv")
harTestdatFilter<- harTestdat[,colSums(is.na(harTestdat))==0]
dim(harTestdatFilter)
```

```
## [1] 20 60
```

```
testingVAl= harTestdatFilter[,c(8:60)]
dim(testingVAl)
```

```
## [1] 20 53
```

```
load("my_model1.rda")
pd = predict(plsFit,newdata=testingVAl)
pd
```

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

Conclusion

partial least squares technique was used for prectiong the classe. with overall accuracy of 61% the confusion matrix is as shown above. The predicted 20 values on the give test dataset are aslo shown above.