PML_Project

vh

June 3, 2017

Project Background and Objectives

The objective of this project is to use machine learning algorithms to predict type of human movement using accelermeter sensor data from 6 participants. These 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). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

Read more: http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz4iysn6fbk (http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz4iysn6fbk)

The goal of this project is to predict the manner in which they did the exercise (Classe variable)

Data Preparation

Data Load

The CSv's are loaded directly using the below code

```
train <- read.csv('pml-training.csv', sep = ",")
test <-read.csv('pml-testing.csv', sep = ",")
```

Data preparation and cleaning

```
train[train == '#DIV/0!'] <- NA
train[train == ''] <- NA
dim(train)
```

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

```
dim(test)
```

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

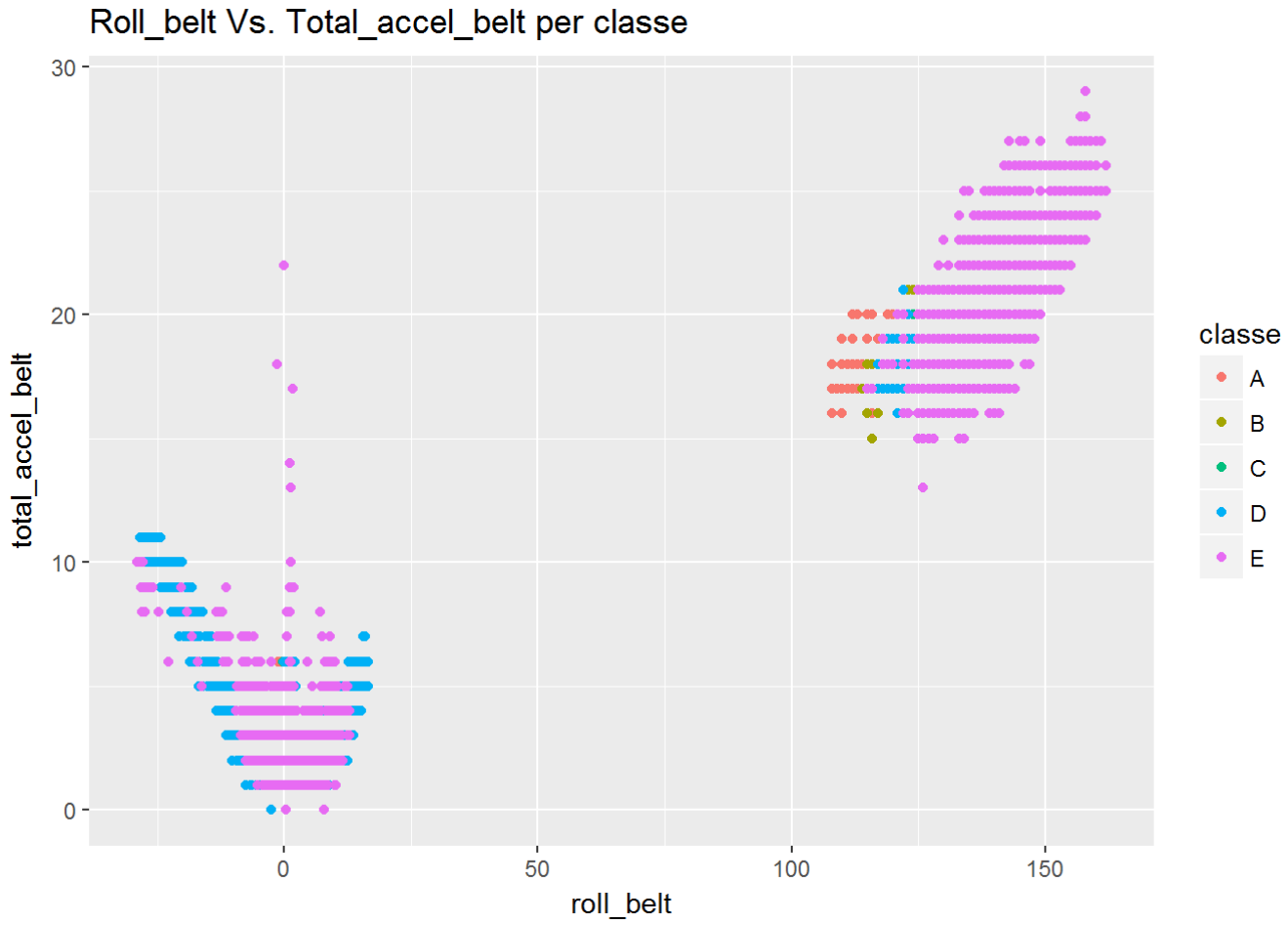
```
train <- train[,-c(1:7)] #removing columns that shouldn't be used in prediction (user name, etc.)
for(i in 1:(ncol(train)-1)){if(class(train[, i]) == 'factor'){train[, i] <- as.numeric(as.character(train[, i]))}} #adjusting datatypes to be numeric
no_var <- nearZeroVar(train, saveMetrics = T) #removing columns with minimal variance
removed.cols <- names(train)[no_var$nzv]
train <- train[,!(no_var$nzv)]
```

Note above, that the dimensions (rows x columns) of the training and testdatsets have been logged. A validation datset will be carved out of the training dataset

Data visualization

Based on an analysis of coorrelations between predictors, we see that are pairs of predictors that havevery high correlation as shown below.

```
qplot(roll_belt, total_accel_belt, data=train, color=classe, main='Roll_belt Vs. Total_accel_belt per classe')
```



Since we will use an ML algorithm that can handle these correlations, we will not remove such predictors.

Machine Learning

We will use Random forest algorithm to build the model. As shown below the model performs well on the training and cross-validation dataset

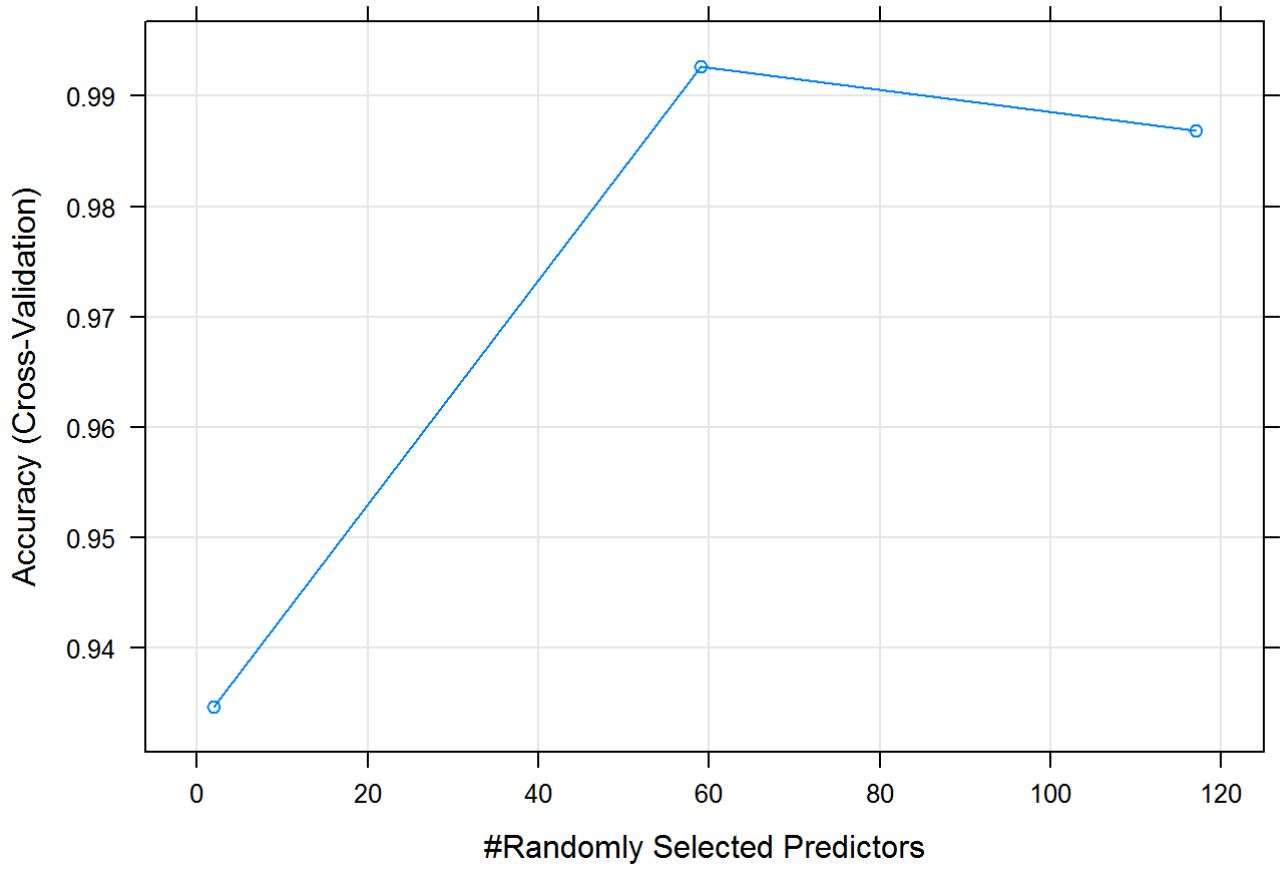
```
library(caret)
train_2 <- train
pr <- preProcess(train_2[,-ncol(train_2)], method="medianImpute")
train_2 <- predict(pr, train_2)
```

```
#parallelizing computations to improve speed
registerDoParallel(cores=4)

cv <- trainControl('cv', 3, savePred=T) #3 fold cross validation
set.seed(111)
model <- train(classe ~ ., data = train_2, method = 'rf', trControl = cv)
model
```

```
## Random Forest
##
## 19622 samples
##   117 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 13081, 13082, 13081
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##     2   0.9346141  0.9171845
##    59   0.9927122  0.9907813
##   117   0.9869025  0.9834310
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 59.
```

```
plot(model)
```



Testing

We now use the model built to test its performance on the test dataset

```
test <- test[,names(test) %in% names(train_2)]
test_2 <- predict(pr, test)
predict(model, newdata=test_2)
```

```
##  [1] 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
```