Weight Lifting Exercise Machine Learning Project

Javier A Diaz

3 de mayo de 2018

Problem

The goal of the project is to predict the manner in which 6 participants did the exercise of weight lifting (EWL). They perform the EWL correctly or incorrectly in 5 ways. The data is obtained from accelerometers on the belt, forearm, arm, and dumbbell, x, y and z axes and 3 Euclidean coordinates for each accelerometer.

DATA

To built the model there are two data sets: training (plm-training) and test (plm-test). First set has 19622 observations with 160 variables. The test set has 20 test with 160 variables.

To have a tidy data,it is preprocess to eliminate missing and undefined (divs by 0) values. The first 7 columns are not relevant variables. A training data frame has 19622 observations and 53 variables.An alternative matrix can be obtain by using the variable window as yes. This matrix have 406 observations with 153 variables. The data with larger number of observations was selected for fitting the model. Model selection was done following steps: feature selection, Cross validation definition, training was done with 4 methods for classification:lda,qda, rf and rpart , selecting the method with accuracy criteria and predict with test set.Results are shown by model at the end.

Features Selection

The outcome variable is Classe. That is a factor with 5 classification levels of performing. Variables with high correlation are identify with the R function findCorrelation ().30 variables was selected. Near zero variables was checked with nearZervar() R function. There is not near Zero variables. An additional, importance of variables were checked but they were of similar importance then the features are the same 30.

Cross Validation

Cross validation with k=10 was used. A test with k=5 was also used but high accuracy is with k=10. CV was done though the R function trainControl() with 10-folds

Accuracy and Model selection

The model with higher accuracy is the Random Forest,rf model 2 the value is 0.9898657 with SD 0.0021 with parameter mtry=2.See Graph 1. OOB estimated error rate of 0.82% with class error lower than 2%.The accuracy is pay with time. Elapsed time, system and user,was 53 min, compared with modelFit3 qda 8 sec.but lower accuracy. Since this is with train data the accuracy could be lower with test data. The second higher model was the Quadratic Dynamic Analysis, qda model 3.the accuracy is 0.805 with SD 0.01 Other model has lower accuracy than .6. Model rf is selected

Prediction

Using the fit with model modFit2 and the Dtest data the prediction for 20 sample is in the predrf file with results shown at the end of the script for model with methods rf and qla

```
library(e1071)
library(lattice)
library(ggplot2)
library(caret)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
DTrain<-read.csv(file="./pml-training.csv")
Dtest<-read.csv("./pml-testing.csv")
div0<-unique(which ( DTrain =='#DIV/0!',arr.ind=TRUE)[,2])
Dtrain<-DTrain[,-div0]

win<-unique(which(is.na(Dtrain),arr.ind=TRUE)[,2]) # columns with missing data are
taken out
Dtrain<-Dtrain[,-win]
##Dtest<-Dtest[,-win]
Dtrain<-Dtrain[,-c(1:7)]
Dtest<-Dtest[,-c(1:7)]

#High corraleted (>,7) are taken out
mc<-findCorrelation(cor(Dtrain[,-53]),cutoff = .7)
Dtrain<-(Dtrain[,-mc])

# Class of integer predictors are change to numeric
for(i in c(1:ncol(Dtrain))) {if(class(Dtrain[,i])=="integer")
  Dtrain[,i]<-as.double(Dtrain[,i])}

# Near zero check
which((nearZeroVar(Dtrain,saveMetrics = T)$nzv)==TRUE)
```

```
## integer(0)
```

```
# Training model1 linear dinamic analysis with cross validation for k=10
set.seed(1234)
Control<-trainControl(method = "cv",number=10)
modFit1<-train(classe~ .,
               method="lda",
               data=Dtrain,trControl=Control)

#Training Model2 Random Forest and with CV and k=10

set.seed(1234)
rfControl<-trainControl(method="cv",number = 10)
modFit2<-train(classe~ .,
               method="rf",data=Dtrain,
               trControl=rfControl)


#training Model 3 with quadratic dinamic analysis method with CV with k=10
set.seed(1234)
qdaControl<-trainControl(method="cv",number=10 )
modFit3<-train(classe~.,method="qda",data=Dtrain,
               verbose=FALSE,size=c(10,20,30),
               trControl=qdaControl)

#training model 4 Recursive Partitioning and Regression tree
set.seed(1234)
tControl<-trainControl(method="cv",number=10 )
modFit4<-train(classe~.,
               method="rpart",data=Dtrain,
               trControl=tControl)
```

MODEL1 lda

```
modFit1$results
```

```
##   parameter  Accuracy     Kappa AccuracySD    KappaSD
## 1      none 0.5850583 0.4743638 0.01078041 0.01360912
```

Model2 rf

```
print(modFit2$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.82%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5574    5    0    1    0 0.001075269
## B   31 3757    9    0    0 0.010534633
## C    1   24 3383   14    0 0.011396844
## D    0    0   62 3150    4 0.020522388
## E    0    0    1    8 3598 0.002495148
```

```
modFit2$results #Selected model with mtry=2
```

```
##   mtry  Accuracy     Kappa  AccuracySD     KappaSD
## 1    2 0.9901649 0.9875571 0.002185507 0.002764708
## 2   16 0.9885335 0.9854934 0.002216267 0.002804794
## 3   30 0.9811431 0.9761423 0.003199697 0.004049358
```

```
print(modFit2$times$'everything') # Time duration
```

```
##    user  system elapsed
## 3294.76   20.87 3334.09
```

```
predrf<-predict(modFit2,Dtest)
predrf
```

```
##  [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
```

Model 3 qla

```
modFit3$results
```

```
##   parameter  Accuracy    Kappa AccuracySD    KappaSD
## 1      none 0.8048641 0.754212 0.01115234 0.01410608
```

```
print(modFit3$times$'everything')
```

```
##    user  system elapsed
##    4.50    0.77    5.32
```

```
predQla<-predict(modFit3,Dtest)
predQla
```

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

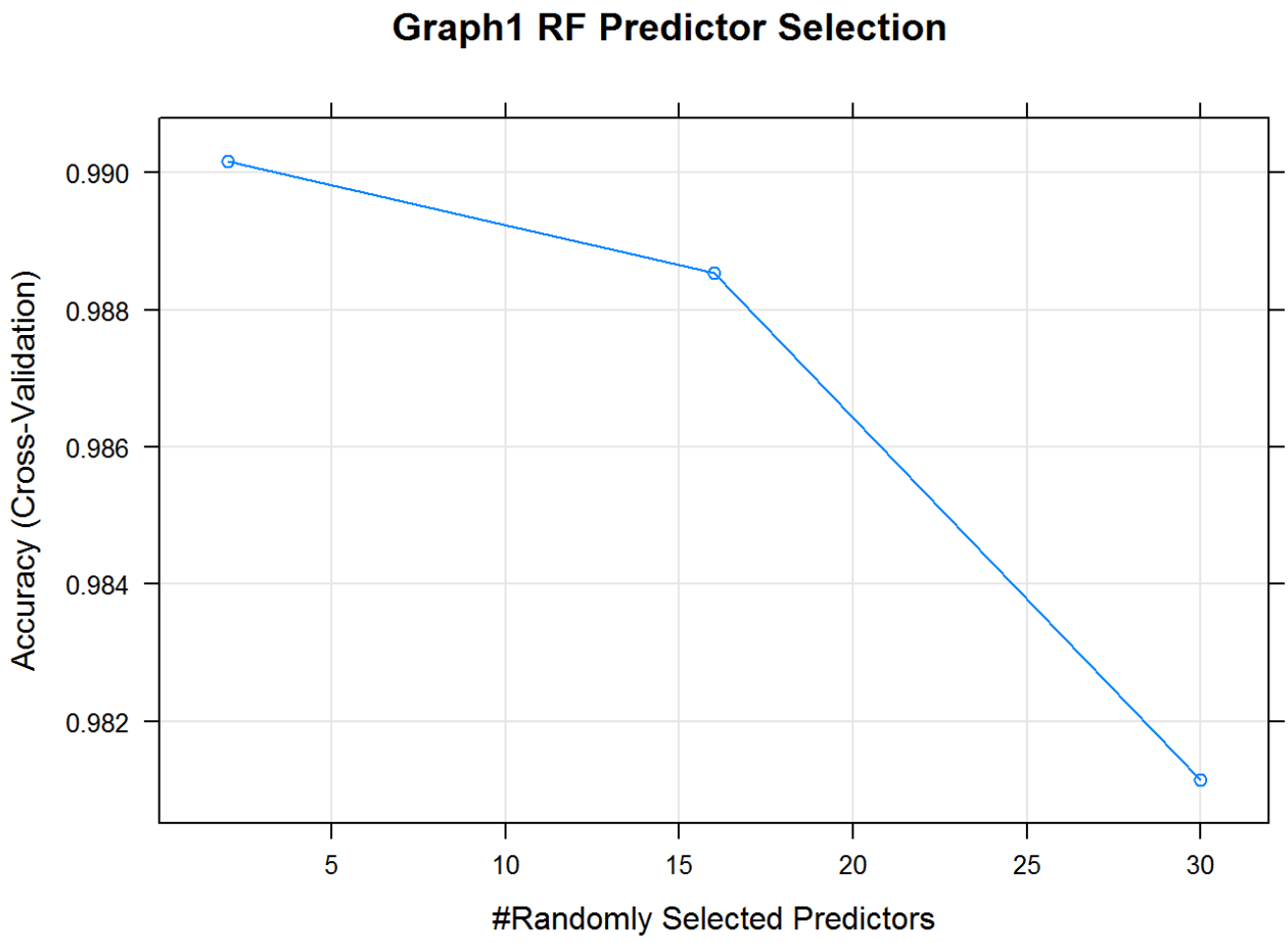
Model 4 rpart

```
modFit4$results
```

```
##           cp  Accuracy     Kappa AccuracySD    KappaSD
## 1 0.02047429 0.5293979 0.3946611 0.03127657 0.04662494
## 2 0.02079476 0.5216037 0.3835397 0.03481061 0.05114414
## 3 0.04226606 0.3945124 0.1829767 0.09976712 0.16414228
```

``` ## Graphs

```
plot(modFit2,type= c("g","o"),main = "Graph1 RF Predictor Selection")
```


```