PML - Course Project

Diego Taccone

24/7/2020

Summary

The objective of the project is to predict the manner in which a series of 20 subjects performed an exercise.

The outcome is the "classe" variable, which classifies if the exercice was performed in either of five different fasions: Class A, B, C, D or E.

We will analyze the Weight Liftting Excersices dataset, create a predictive model with the training dataset, and test its out of sample error on the test data set.

Data Download and Preparation

We download and clean the data. Variables without values and NA´s are eliminated from the dataset, both in the training and the testing sets.

```
data <- read.csv("training.csv")
testing <- read.csv("testing.csv")
dataclean <- data[,!is.na(data[1,])]
dataclean <- dataclean[,dataclean[1,] != ""]
dataclean$classe <- as.factor(dataclean$classe)
testing <- testing[,!is.na(testing[1,])]
```

Analysis

Being a classification problem, the approach will be first the check de accuracy of a simple classification tree, then a Random Forest.

For this, we will subset the training set, we called **data** into a training and testing set, being 70% training and 30% testing. We will train the model with this sub training set, then measure the accuracy on the sub testing set.

The training function in the caret package will be used, and will be set to perform a 25 Bootstrap Cross Validation.

```
library(caret)
```

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

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

```
inTrain <- createDataPartition(dataclean$classe,p=0.7,list = FALSE)
training <- dataclean[inTrain,]
testset <- dataclean[-inTrain,]
modFit <- train(classe~.,method = "rpart",data = training[,-c(1,2)])
modFit$resample
```

```
##     Accuracy     Kappa   Resample
## 1  0.3566766 0.1245903 Resample09
## 2  0.3801076 0.1370743 Resample05
## 3  0.3734678 0.1272762 Resample01
## 4  0.5571202 0.4189652 Resample10
## 5  0.5380630 0.4172973 Resample06
## 6  0.3662698 0.1213185 Resample02
## 7  0.3584643 0.1192822 Resample11
## 8  0.3740322 0.1355410 Resample07
## 9  0.3614791 0.1237663 Resample03
## 10 0.6030369 0.4973194 Resample12
## 11 0.3739249 0.1247832 Resample08
## 12 0.5016983 0.3575644 Resample04
## 13 0.3680402 0.1248160 Resample13
## 14 0.4908184 0.3329212 Resample22
## 15 0.3662281 0.1256157 Resample18
## 16 0.3613546 0.1272992 Resample14
## 17 0.3679937 0.1254581 Resample23
## 18 0.4973822 0.3683977 Resample19
## 19 0.3649980 0.1211940 Resample15
## 20 0.4925462 0.3644151 Resample24
## 21 0.5658517 0.4530270 Resample20
## 22 0.4663899 0.3123774 Resample16
## 23 0.3602559 0.1203630 Resample25
## 24 0.3698388 0.1257284 Resample21
## 25 0.4695532 0.3097724 Resample17
```

In this first model, we see that the Accuracy is very low.

We now try with Random Forests and check te accuracy on the Bootstrap Samples.

```
modFit1 <- train(classe~.,method = "rf",data = training[,-c(1,2)])
modFit1$resample
```

```
##     Accuracy     Kappa   Resample
## 1  0.9992103 0.9990004 Resample04
## 2  0.9982164 0.9977460 Resample09
## 3  0.9994072 0.9992502 Resample05
## 4  0.9980214 0.9974977 Resample01
## 5  0.9988064 0.9984886 Resample10
## 6  0.9983994 0.9979759 Resample06
## 7  0.9974201 0.9967366 Resample02
## 8  0.9974425 0.9967642 Resample11
## 9  0.9988050 0.9984922 Resample07
## 10 0.9984289 0.9980128 Resample03
## 11 0.9976285 0.9970045 Resample12
## 12 0.9972354 0.9965031 Resample08
```

```
## 13 0.9974470 0.9967698 Resample17
## 14 0.9984038 0.9979780 Resample13
## 15 0.9978317 0.9972582 Resample22
## 16 0.9976091 0.9969687 Resample18
## 17 0.9982171 0.9977455 Resample14
## 18 0.9982075 0.9977380 Resample23
## 19 0.9974191 0.9967350 Resample19
## 20 0.9984076 0.9979878 Resample15
## 21 0.9982185 0.9977498 Resample24
## 22 0.9980354 0.9975164 Resample20
## 23 0.9988038 0.9984841 Resample16
## 24 0.9980123 0.9974888 Resample25
## 25 0.9988034 0.9984854 Resample21
```

We can observe on the results that we have a very high accuracy on trained model.

We check accuracy with the test set.

```
predTest <- predict(modFit1,testset[,-c(1,2)])
confusionMatrix(predTest,testset$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1139    2    0    0
##          C    0    0 1024    0    0
##          D    0    0    0  964    0
##          E    0    0    0    0 1082
##
## Overall Statistics
##
##                Accuracy : 0.9997
##                  95% CI : (0.9988, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9996
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   0.9981   1.0000   1.0000
## Specificity            1.0000   0.9996   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   0.9982   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   0.9996   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1935   0.1740   0.1638   0.1839
## Detection Prevalence   0.2845   0.1939   0.1740   0.1638   0.1839
## Balanced Accuracy      1.0000   0.9998   0.9990   1.0000   1.0000
```

Observing the values of the confusionMatrix, we can see the very high accuracy this Random Forest model achieves on our test set.

Prediction of 20 samples

Now we have selected our model, we can predict our values for the 20 samples.

```
predVal <- predict(modFit1,testing[,-c(1,2,60)])
predVal
```

```
##  [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
```

These predicted values where 100% percent certain in the prediction quiz asociated with the project.