Practical Machine Learning Project - Prediction

Francisco J. Jaramillo

Friday, February 20, 2015

1.- DEVELOPING THE PREDICTION MODEL

Different methods and packages were tried to build the model. It was decided to use the package caret for partitioning; and develop a model based on the random Forest method using the randomForrest package.

1.1- Loading the data

```
    library(caret)
```

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

```
    setwd("C:/Users/Francisco/Desktop/DDatasScience/08 - MachLearn/Project")
    data = read.csv("pml-training.csv")
```

1.2- Selecting relevant columns

The original data set has 160 columns. Some of the columns are detail data from the measurement devices while other columns are subtotals and totals. The data for prediction is based in the detail from the measurement devices. So, columns containing data from totals and subtotals was ignored; as they are not relevant for this specific project. The following vector “col” include only relevant columns.

```
    col<-c(2,8,9,10,11,37,38,39,40,41,42,43,44,45,46,47,48,49,60,61,62,63,64
           ,65,66,67,68,84,85,86,102,113,114,115,116,117,118,119,120,121,122
           ,123,124,140,151,152,153,154,155,156,157,158,159,160)
```

1.3- Removing no relevant rows

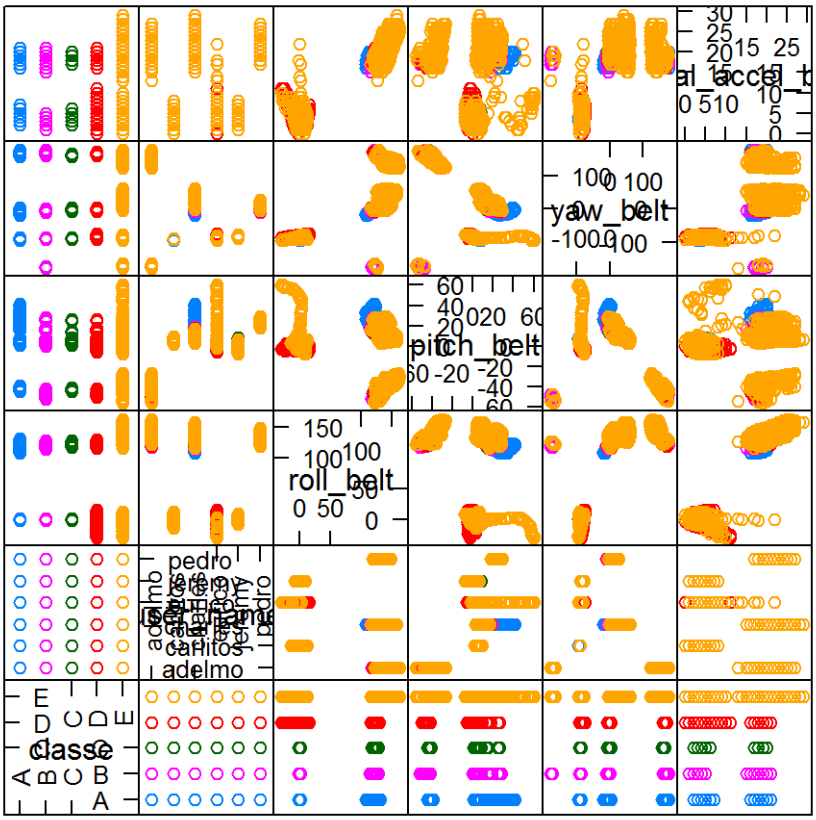
rdata is a subset of data after non relevant columns have been removed.

```
    rdata<-data[ ,col]
```

1.4- Select variables with the highest explanatory potential.

Started with 160 columns of data. Removed 106 of subtotals considered irrelevant, remaining 54 columns. But the chosen methodology for the development of the model was to identify the smallest data set that produced a model with good accuracy. For that purpose, the 54 columns remaining were plotted against the independent variable “classe”. The goal was to identify by visual observation of the graphs, which are the columns/variables which represent the co-variants with higher explanatory potential. The 54 columns were plotted on 11 graphs, and 13 columns were selected as showing the highest explanatory potential. For the purpose of this final submission only one plot will be included in this write-up as a sample.

```
    featurePlot(x=rdata[,c(54,1,2,3,4,5)],y=rdata$classe,plot="pairs")
```



Scatter Plot Matrix

```
#featurePlot(x=rdata[,c(54,6,7,8,9,10)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,11,12,13,14,15)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,16,17,18,19,20)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,21,22,23,24,25)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,26,27,28,29,30)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,31,32,33,34,35)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,36,37,38,39,40)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,41,42,43,44,45)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,46,47,48,49,50)],y=rDataU1$classe,plot="pairs")
#featurePlot(x=rdata[,c(54,51,52)],y=rDataU1$classe,plot="pairs")
```

1.5- Columns selected with highest explanatory power

The following vector dset contains the 13 columns considered to have the highest explanatory potential

```
dset<-c(54,1,2,3,15,29,30,35,36,37,38,45,52,53)
```

1.6- Keeping selected columns

```
sdata<-rdata[ ,dset]
```

1.7- Partitioning the training dataset for cross validation

In order to apply some cross validation the training data was separated in two partitions 90% for training of the model Vs 10% for cross validation.

```
set.seed(777)
inTrain <- createDataPartition(y=sdata$classe, p=0.9, list=FALSE)
training <- sdata[inTrain,]
testing <- sdata[-inTrain,]
```

1.8- Obtaining randomForest prediction model

Package “randomForest”" was used to train the data and build the model. In a preliminar step “caret”" was used to build a random forest model. but “caret” turned out to be much more demanding on computer resources than “randomForest”.

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
rf = randomForest(training$classe~., data=training, mtry=4, ntree=100, importance=TRUE)
print(rf)
```

```
##
## Call:
##  randomForest(formula = training$classe ~ ., data = training,      mtry = 4, ntree = 100, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 2.96%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4927   20   35   34    6     0.01892
## B   56 3281   59   18    4     0.04008
## C   40   34 2963   39    4     0.03799
## D   37   10   62 2784    2     0.03834
## E    8   19   17   18 3185     0.01909
```

Results:

Model rf was created using the random forrest model for a partition of 17662 rows

According to the print out of the model, the error is expected to be about 2.96%

2.- TESTING/CROSSVALIDATION

2.1- Predicting

Here the model is used to predict the “classe” (A,B,C,D,E) of the testing partition and compare the result to the know “classe”" using a Confusion Matrix

```
pred<-predict(rf,testing)
confusionMatrix(pred,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 545  10   3   4   2
##          B   5 354   1   0   4
##          C   6  10 334   3   1
##          D   2   4   3 314   3
##          E   0   1   1   0 350
##
## Overall Statistics
##
##                Accuracy : 0.968
##                  95% CI : (0.959, 0.975)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.959
##  Mcnemar's Test P-Value : 0.0179
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.977    0.934    0.977    0.978    0.972
## Specificity             0.986    0.994    0.988    0.993    0.999
## Pos Pred Value          0.966    0.973    0.944    0.963    0.994
## Neg Pred Value          0.991    0.984    0.995    0.996    0.994
## Prevalence              0.285    0.193    0.174    0.164    0.184
## Detection Rate          0.278    0.181    0.170    0.160    0.179
## Detection Prevalence    0.288    0.186    0.181    0.166    0.180
## Balanced Accuracy       0.982    0.964    0.982    0.985    0.985
```

Results:

According to the confusion Matrix on the crossvalidation the calculated accuracy is 96.8%

while the interval of confidence at 95% calculates the accuraccy should be between 95.9% and 97.5%

Interesting Note:

The calculated prediction model matched the results of the exercise “prediction assignment submission”

with 100% accuraccy.
