Prediction Assignment Writeup

Oscar Lado Baleato

31th July of 2016

Summary

The methods of classification are so important nowadays in different fields, medicine, biology, business... In this exercise I will try to classify the way of doing exercicise in five different categories (A,B,C,D,E).

To do so, we have two data sets, one to build the model and the other to predict the class of a new data.

Loading, preparing and cleaning the data

First, we get the data and remove the NAs variables, and variables without useful information.

```
setwd("C:/datos")
data <- read.csv("data_train.csv", na.strings = c("NA", "#DIV/0!", ""))
testin<-read.csv("data_test.csv",na.strings=c("NA","#DIV/0!"))

NA_Count = sapply(1:dim(data)[2],function(x)sum(is.na(data[,x])))
NA_list = which(NA_Count>0)

data = data[,-NA_list]
data = data[,-c(1:7)]
data$classe = factor(data$classe)

NA_Count1 = sapply(1:dim(testin)[2],function(x)sum(is.na(testin[,x])))
NA_list1 = which(NA_Count1>0)
testing = testin[,-NA_list]
testing = testin[,-c(1:7)]
dim(data)
```

```
## [1] 19622    53
```

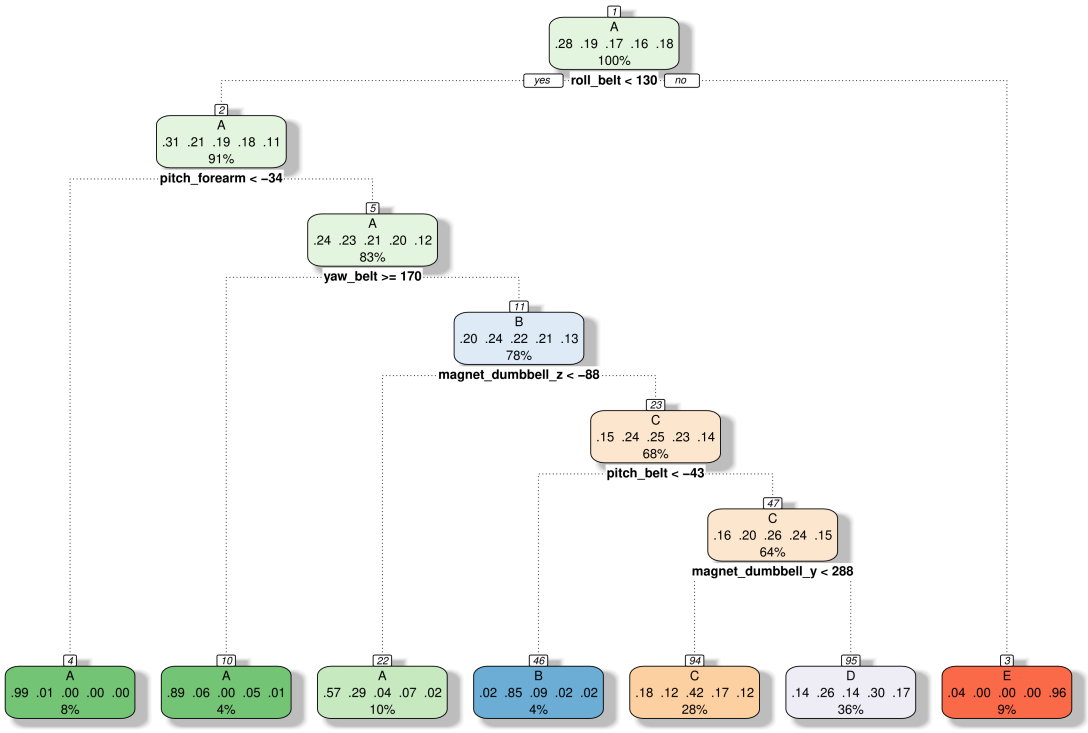
To create the model, we first create the training and testing datasets.

```
inTrain=createDataPartition(y=data$classe, p=0.6, list=FALSE)
training <-data[inTrain,]
testing <- data[-inTrain,]
```

Create the model

I classify the data with a classification tree, and then I use cross validation with the same method to improve the classification.

```
model1<- train(classe ~ .,method='rpart',data=training)
fancyRpartPlot(model1$finalModel)
```



Rattle 2016−xul−31 21:42:09 Óscar

```
pred=predict(model1,newdata=testing)
z=confusionMatrix(pred,testing$classe)
z$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1330  240   47   83   20
##          B    4  261   32    9    8
##          C  413  259  887  359  252
##          D  452  758  402  835  469
##          E   33    0    0    0  693
```

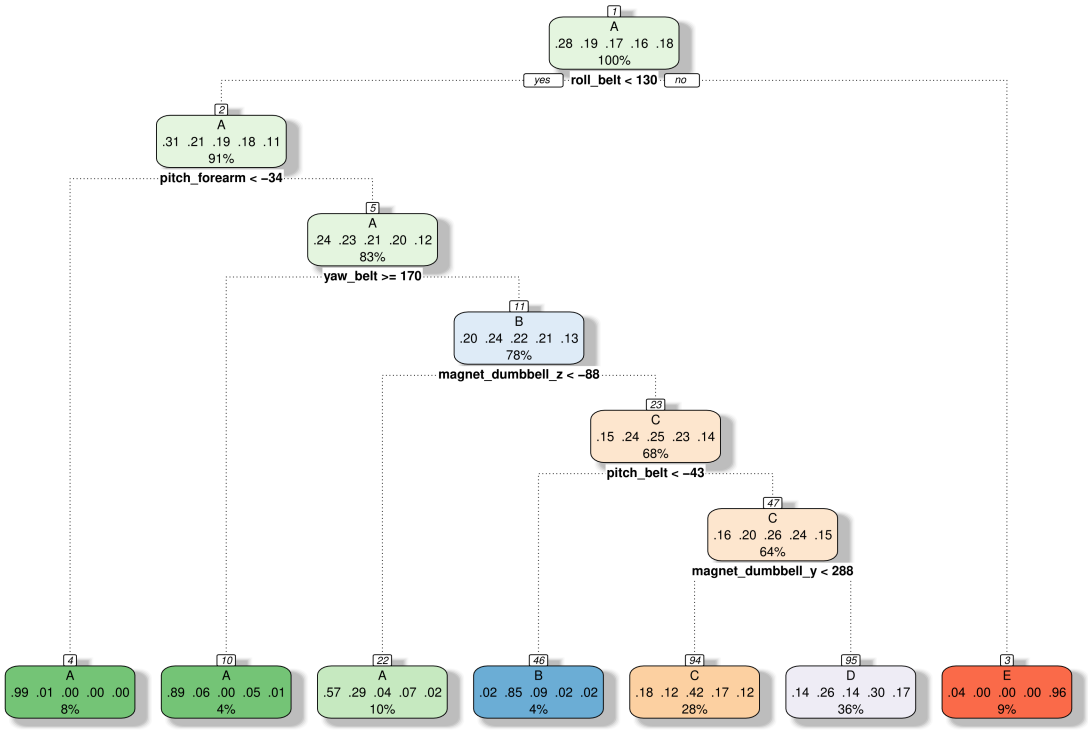
```
z$overall[1]
```

```
##  Accuracy
## 0.5105786
```

From the confusion matrix it is clear the accuracy of "0.49" which is the same than a random classification, so it is no a good model to this data

We fit the model again, but using cross validation

```
train_control<- trainControl(method="cv", number=10, savePredictions = TRUE)
model2<- train(classe~., data=training, trControl=train_control, method="rpart")
fancyRpartPlot(model2$finalModel)
```




```
pred=predict(model2,newdata=testing)
z=confusionMatrix(pred,testing$classe)
z$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1330  240   47   83   20
##          B    4  261   32    9    8
##          C  413  259  887  359  252
##          D  452  758  402  835  469
##          E   33    0    0    0  693
```

```
z$overall[1]
```

```
##  Accuracy
## 0.5105786
```

We get the same accuracy than in the previous model.

Random Forest Method

We use now random forest, with and without cross validation, to get a better accuracy of classification.

```
model3=randomForest(classe~., data=training, method='class')
pred=predict(model3,testing,type='class')
```

```
z2=confusionMatrix(pred,testing$classe)
z2$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 2230   15    0    0    0
##          B    1 1498   11    0    0
##          C    0    5 1356   29    1
##          D    0    0    1 1257    2
##          E    1    0    0    0 1439
```

```
z2$overall[1]
```

```
##  Accuracy
## 0.9915881
```

This model provides 99% accurancy hence this model has been choosen to do predict the testing data set. We can check the useful of cross validation to improve the model before go to the new dataset.

```
train_control<- trainControl(method="cv", number=10, savePredictions = TRUE)
model4=randomForest(classe~.,trControl=train_control,data=training, method='class')
```

```
pred= predict(model4,testing,type='class')
z2=confusionMatrix(pred,testing$classe)
z2$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 2229   12    0    0    0
##          B    1 1501   11    0    0
##          C    0    5 1356   30    0
##          D    1    0    1 1255    2
##          E    1    0    0    1 1440
```

```
z2$overall[1]
```

```
##  Accuracy
## 0.9917155
```

Conclusion

I can conclude that for this data, the best model is a random forest, and the cross validation do not give to us a better fit.

Solution to the FINAL QUIZ

```
prediction=predict(model3,testin,type='class')
nofiles = length(prediction)
for (i in 1:nofiles){
  filename =  paste0("problem_id",i,".txt")
  write.table(prediction[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
}
prediction
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```