PRACTICAL MACHINE LEARNING

ASSIGMENT WEEK 3

# Steps

1. ***Row Data***

After a first to the row data, some important problems arise:

* On certain columns there are NAs, empty value and ‘#DIV/0!’ strings (probably the informations was dumped from excel. To solve this problem we use the following argument in the ‘read.csv’ function 🡪 na.string=c("NA","NaN","","#DIV/0!")

data<-read.csv("pml-training.csv",sep = ",",dec=".",na.string=c("NA","NaN","","#DIV/0!"))

* There are some columns with an important number of NAs or empty value. It would be convenient with take them out since the number of columns is huge (160) and since these column do not give a lot of information. To do so, we create a function to ‘detect’ this columns named ‘nullcol’.

nullcol <- function (x,thresh) {

nullmatrix<-is.na(x)+is.null(x)

numnulls<-apply(nullmatrix,2,sum)

dim(numnulls)<-c(1,dim(x)[2])

nullcol<-(numnulls/dim(x)[1]>thresh)

}

data\_light<-data[,nullcol(data,0.8)==FALSE]

1. ***Training and Testings data sets***

In order to do ***cross-validation*** we devide the information contained in the ‘training’ csv into training and testing data set. Since the numbers of rows is wide, it is possible to create training and testing data sets with a significant number of rows. We use 75% of the rows for the training data set.

(y=data\_light$classe,p=0.75,list=FALSE)

training<-data\_light[inTrain,]

testing<-data\_light[-inTrain,]

1. ***Preprocessing with PCA***

The numbers of columns seems to be too high, so we try preprocessing. Previously we check the correlation of the different columns with *cor* function and see that there are an important numbers of pairs of columns with a correlation higher than 80%. We decide to use PCA as a ‘clean’ method for preprocessing. Since the accelerometers were place in 4 places (belt, forearm, arm, and dumbbell), 4 seems to be a minimum for the number of componets. Additionally, we take all the columns that are of class factor or that haven´t usefull information.

cols\_out<-c(-1,-2,-3,-4,-5,-6,-7,-60)

preProc<-preProcess(training[,cols\_out],method="pca",pcaComp=4)

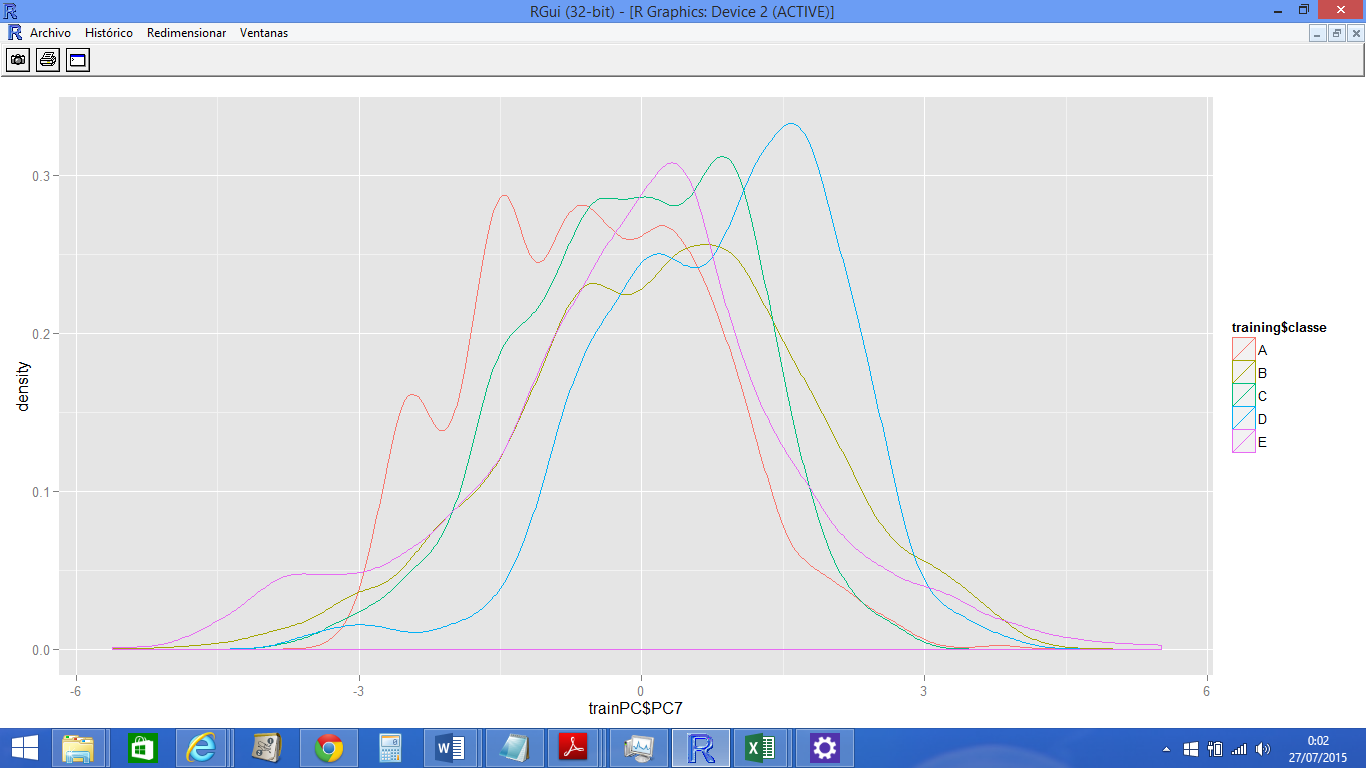
trainPC <- predict(preProc,training[,cols\_out])

testPC <- predict(preProc,testing[,cols\_out])

1. ***Plotting***

Drawing the histogram of the components in the different classe’s (in different colours), it is possible to see some characteristic patterns of each classe. However, it is impossible to reach any conclusion with a single component.

qplot(trainPC$PC7,colour=training$classe,geom="density")



1. ***Selecting the model and launching train***

Since the result (classe) is a factor (A,B,C,D or E), linear models does not seem to be usefull. We try with a tree, but the accuracy is really poor (<40%). So, finally we decide to try with a Random Foster. We launch it and go to do some shopping since it is really slow … ;)

modelFit<-train(training$classe ~ .,method="rf",data=trainPC)

1. ***Prediction and Accuracy***

We have to use PCA in the testing set too. We predict an check the accuracy with confusionMatrix. It is higher than 90%. Hurray!!!!. With this result we can assure that the **out of sample error** is lower than 10%.

testPC <- predict(preProc,testing[,cols\_out])

testpredict<-predict(modelFit,testPC)

confusionMatrix(testing$classe,testpredict)

***Confusion Matrix and Statistics***

***Reference***

***Prediction A B C D E***

***A 1340 22 19 10 4***

***B 20 870 35 17 7***

***C 23 17 801 10 4***

***D 12 5 40 737 10***

***E 6 8 18 14 855***

***Overall Statistics***

***Accuracy : 0.9386***

***95% CI : (0.9315, 0.9452)***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****