Machine Learning

Vinsent

Monday, April 4, 2016

### Prediction Assignment1

### Overview:

### The goal of your project is to predict the manner in which they did the exercise by using any of the machine learning model. Then evaluate the results obtained by cross validating the prediction and actual results. The model also needs to validate the given 20 test cases and the error in predicting them.

### 

## Read the training data

# load the libraries  
library(class)  
library(gmodels)  
library(corrgram)

## Warning: package 'corrgram' was built under R version 3.2.4

library(ellipse)  
  
  
as1\_train <- read.csv("G:/coursera/DataScience\_specialization/practical machine learning/assignment1/pml-training\_01.csv", header = TRUE, sep = ",")  
  
# summary(as1\_train)   
  
# summary(as1\_train$classe)

## Read the test data and its summary

as1\_test <- read.csv("G:/coursera/DataScience\_specialization/practical machine learning/assignment1/pml-training\_01.csv", header = TRUE, sep = ",")  
# summary(as1\_test)

# PreProcessing

## Check the number of target classes. Plot these values

## PCA principle component analysis to reduce the inputs to the model.

#### removing the time stamp columns to get rid of the dates column for PCA...

#### removed the NA, completely missing value columns for PCA input.

#### selecting only a subset of columns from the original data set as1\_train

#### the below subset command did not work.

#### as1\_test.sub2 <- as1\_test[, c(user\_name, num\_window,roll\_belt,pitch\_belt,yaw\_belt,total\_accel\_belt,kurtosis\_roll######\_beearm\_y,magnet\_forearm\_z,problem\_id)]

##### as1.test.sub2 <- as1\_test[, c(1,2)]

##### x.sub2 <- subset(as1\_test, select = c(1,3,4))

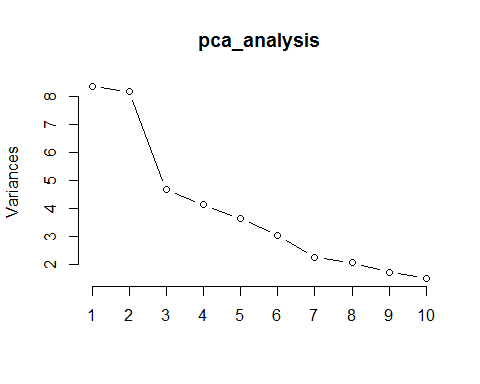
as1\_train <- read.csv("G:/coursera/DataScience\_specialization/practical machine learning/assignment1/pml-training\_01.csv", header = TRUE, sep = ",")  
  
#### after cleaning the dataset we have around 54 columns from 159 columns..   
#### the last variable is the target class  
ncol(as1\_train)

## [1] 55

pca\_analysis <- prcomp(as1\_train[1:54], scale.=TRUE)  
  
#!summary(pca\_analysis)  
  
sum((pca\_analysis$sdev)^2)

## [1] 54

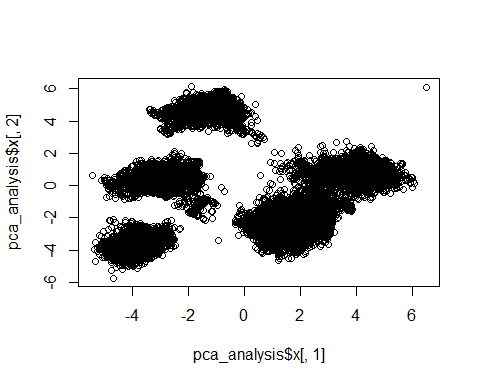
screeplot(pca\_analysis, type="lines")



(pca\_analysis$sdev)^2

## [1] 8.356570172 8.157815849 4.680516775 4.129791796 3.653030206  
## [6] 3.032841966 2.251446042 2.074181484 1.725702211 1.509380865  
## [11] 1.396554772 1.152153183 1.045158291 0.998367113 0.944813173  
## [16] 0.885620566 0.805433462 0.727531035 0.677386427 0.600709013  
## [21] 0.528925794 0.481039208 0.417822070 0.389818048 0.382460056  
## [26] 0.334236645 0.305830962 0.290899406 0.255355082 0.233671195  
## [31] 0.203412168 0.179755274 0.169989400 0.131132895 0.121759668  
## [36] 0.112181282 0.091890739 0.079717372 0.063952948 0.056406421  
## [41] 0.055131359 0.040797172 0.037730038 0.035291857 0.033662078  
## [46] 0.031450611 0.028617467 0.026551884 0.021661870 0.020426383  
## [51] 0.013439953 0.011874638 0.005954931 0.002148724

# PCA 1 to 14 can be combined to have 0.81 % of variation in the dataset. but for this study I am taking only PCA1 and PCA 2. Also this is able to capure the five groups A,B,C,D,E  
plot(pca\_analysis$x[,1],pca\_analysis$x[,2])  
text(pca\_analysis$x[,1],pca\_analysis$x[,2], pca\_analysis$V1, cex=0.7, pos=4, col="red")



pca\_analysis$rotation[,1]

## new\_window num\_window roll\_belt   
## -2.349418e-03 2.892623e-03 -3.067589e-01   
## pitch\_belt yaw\_belt total\_accel\_belt   
## -2.595569e-02 -1.993288e-01 -3.034154e-01   
## gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z   
## 9.571321e-02 -1.018383e-01 1.799252e-01   
## accel\_belt\_x accel\_belt\_y accel\_belt\_z   
## 1.089242e-02 -3.165177e-01 3.158463e-01   
## magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z   
## -1.424278e-02 1.170732e-01 6.049548e-02   
## roll\_arm pitch\_arm yaw\_arm   
## 6.152848e-02 3.703377e-02 5.008837e-02   
## total\_accel\_arm gyros\_arm\_x gyros\_arm\_y   
## 1.109588e-01 -1.094312e-02 7.511105e-02   
## gyros\_arm\_z accel\_arm\_x accel\_arm\_y   
## -1.559444e-01 -1.620109e-01 2.680228e-01   
## accel\_arm\_z magnet\_arm\_x magnet\_arm\_y   
## -1.265057e-01 -9.080482e-02 6.600444e-02   
## magnet\_arm\_z roll\_dumbbell pitch\_dumbbell   
## 3.272818e-02 8.779864e-02 -1.103710e-01   
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x   
## -1.263303e-01 1.694594e-01 -3.457356e-03   
## gyros\_dumbbell\_y gyros\_dumbbell\_z accel\_dumbbell\_x   
## -8.234470e-04 -1.978944e-04 -1.711847e-01   
## accel\_dumbbell\_y accel\_dumbbell\_z magnet\_dumbbell\_x   
## 1.827605e-01 -1.552965e-01 -1.701897e-01   
## magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm   
## 1.470041e-01 1.704761e-01 6.443666e-02   
## pitch\_forearm yaw\_forearm total\_accel\_forearm   
## -1.460158e-01 1.136102e-01 -6.290671e-05   
## gyros\_forearm\_x gyros\_forearm\_y gyros\_forearm\_z   
## -6.850299e-02 -3.349934e-03 2.312228e-03   
## accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z   
## 1.913320e-01 3.553536e-02 -3.103106e-02   
## magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z   
## 1.051439e-01 2.502355e-02 -3.771553e-02

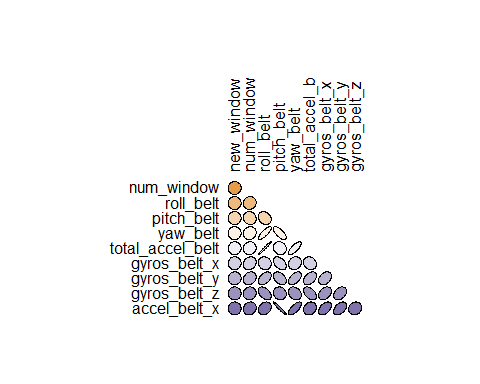
# Exploratory Analysis To see how the variables are correlated.

### The below exploration also explains what the variable highly correlated, This indicates like what variables are good for the prediction. A good feature selection should include uncorrelated variables to have less bias in your prediction.

# corrgram(as1\_train, order= NULL, lower.panel=panel.shade,  
# upper.panel = NULL, text.panel=panel.txt,  
# main="Groupware Human Activity Recognition Data in PC2/PC1 Order")  
# another colored corrgram  
  
R = cor(as1\_train[, 1:10])  
round(R, 3)

## new\_window num\_window roll\_belt pitch\_belt yaw\_belt  
## new\_window 1.000 0.009 0.009 0.001 0.004  
## num\_window 0.009 1.000 0.072 -0.102 0.086  
## roll\_belt 0.009 0.072 1.000 -0.216 0.815  
## pitch\_belt 0.001 -0.102 -0.216 1.000 -0.700  
## yaw\_belt 0.004 0.086 0.815 -0.700 1.000  
## total\_accel\_belt 0.009 0.066 0.981 -0.139 0.762  
## gyros\_belt\_x 0.000 0.210 -0.117 -0.436 0.145  
## gyros\_belt\_y 0.009 0.224 0.464 -0.397 0.530  
## gyros\_belt\_z 0.002 0.067 -0.459 -0.107 -0.275  
## accel\_belt\_x 0.000 0.133 0.257 -0.966 0.708  
## total\_accel\_belt gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z  
## new\_window 0.009 0.000 0.009 0.002  
## num\_window 0.066 0.210 0.224 0.067  
## roll\_belt 0.981 -0.117 0.464 -0.459  
## pitch\_belt -0.139 -0.436 -0.397 -0.107  
## yaw\_belt 0.762 0.145 0.530 -0.275  
## total\_accel\_belt 1.000 -0.165 0.409 -0.475  
## gyros\_belt\_x -0.165 1.000 0.333 0.340  
## gyros\_belt\_y 0.409 0.333 1.000 0.342  
## gyros\_belt\_z -0.475 0.340 0.342 1.000  
## accel\_belt\_x 0.172 0.474 0.447 0.117  
## accel\_belt\_x  
## new\_window 0.000  
## num\_window 0.133  
## roll\_belt 0.257  
## pitch\_belt -0.966  
## yaw\_belt 0.708  
## total\_accel\_belt 0.172  
## gyros\_belt\_x 0.474  
## gyros\_belt\_y 0.447  
## gyros\_belt\_z 0.117  
## accel\_belt\_x 1.000

plotcorr(R, col = colorRampPalette(c("#E08214", "white", "#8073AC"))(10), type = "lower")



# reference http://little-book-of-r-for-multivariate-analysis.readthedocs.org/en/latest/src/multivariateanalysis.html  
  
  
  
# Loading the data to check the proportion in each case A to E. in percentage  
  
round(prop.table(table(as1\_train$classe)) \* 100, digits = 1)

##   
## A B C D E   
## 28.4 19.4 17.4 16.4 18.4

round(prop.table(table(as1\_test$classe)) \* 100, digits = 1)

##   
## A B C D E   
## 28.4 19.4 17.4 16.4 18.4

# Normalizing the data   
  
normalize <- function(x) {  
return ((x - min(x)) / (max(x) - min(x)))  
}  
  
as1\_train\_n <- as.data.frame(lapply(as1\_train[1:54], normalize))  
as1\_test\_n <- as.data.frame(lapply(as1\_test[1:54], normalize))

## KNN Classifier

# Extracting the labels  
as1\_train\_labels <- as1\_train$classe  
as1\_test\_labels <- as1\_test$classe  
  
# Need to remove the missing values in the training and test dataframes  
  
as1\_train\_n1 <- na.omit(as1\_train\_n)  
# as1\_test\_n1 <- na.omit(as1\_test\_n)  
  
# Prediction  
  
as1\_pred <- knn(train = as1\_train\_n1, test = as1\_test\_n, cl = as1\_train\_labels, k=5)  
  
# test set classification  
  
as1\_test\_pred <- knn(train = as1\_train\_n1, test = as1\_test\_n,  
cl = as1\_train\_labels, k=5)  
  
# The below are the predicted lables.  
head(as1\_test\_pred, 10)

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

## Evaluating the model

library(gmodels)  
  
# I have assigned some fictitious data for the label as there were no lables given for the test set.  
  
  
CrossTable(x = as1\_test\_labels, y = as1\_test\_pred,  
prop.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 19622   
##   
##   
## | as1\_test\_pred   
## as1\_test\_labels | A | B | C | D | E | Row Total |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## A | 5536 | 20 | 8 | 11 | 5 | 5580 |   
## | 0.992 | 0.004 | 0.001 | 0.002 | 0.001 | 0.284 |   
## | 0.992 | 0.005 | 0.002 | 0.003 | 0.001 | |   
## | 0.282 | 0.001 | 0.000 | 0.001 | 0.000 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## B | 29 | 3695 | 49 | 15 | 9 | 3797 |   
## | 0.008 | 0.973 | 0.013 | 0.004 | 0.002 | 0.194 |   
## | 0.005 | 0.983 | 0.014 | 0.005 | 0.003 | |   
## | 0.001 | 0.188 | 0.002 | 0.001 | 0.000 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## C | 5 | 22 | 3364 | 27 | 4 | 3422 |   
## | 0.001 | 0.006 | 0.983 | 0.008 | 0.001 | 0.174 |   
## | 0.001 | 0.006 | 0.961 | 0.008 | 0.001 | |   
## | 0.000 | 0.001 | 0.171 | 0.001 | 0.000 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## D | 9 | 8 | 63 | 3127 | 9 | 3216 |   
## | 0.003 | 0.002 | 0.020 | 0.972 | 0.003 | 0.164 |   
## | 0.002 | 0.002 | 0.018 | 0.979 | 0.003 | |   
## | 0.000 | 0.000 | 0.003 | 0.159 | 0.000 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## E | 4 | 13 | 15 | 14 | 3561 | 3607 |   
## | 0.001 | 0.004 | 0.004 | 0.004 | 0.987 | 0.184 |   
## | 0.001 | 0.003 | 0.004 | 0.004 | 0.992 | |   
## | 0.000 | 0.001 | 0.001 | 0.001 | 0.181 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## Column Total | 5583 | 3758 | 3499 | 3194 | 3588 | 19622 |   
## | 0.285 | 0.192 | 0.178 | 0.163 | 0.183 | |   
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|  
##   
##

### The below are the predicted lables.

as1\_test\_pred

#### [1] B A A A A E D B A A B A B A E E E B B B

#### Levels: A B C D E

## Inference from the prediction

### there were no labels for 'C' Category, A was predicted as A 100% classified correctly, There were 10% miss classification ####for 'B', C were all miss classified. D was also calssified correctly 100%. E was also calssified correctly 100%.

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.