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

# 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-
# 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,ku

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

#### after cleaning the dataset we have around 54 columns from 159 columns..
#### the last variable is the target class
ncol(as1_train)
```

```
## [1] 55
```

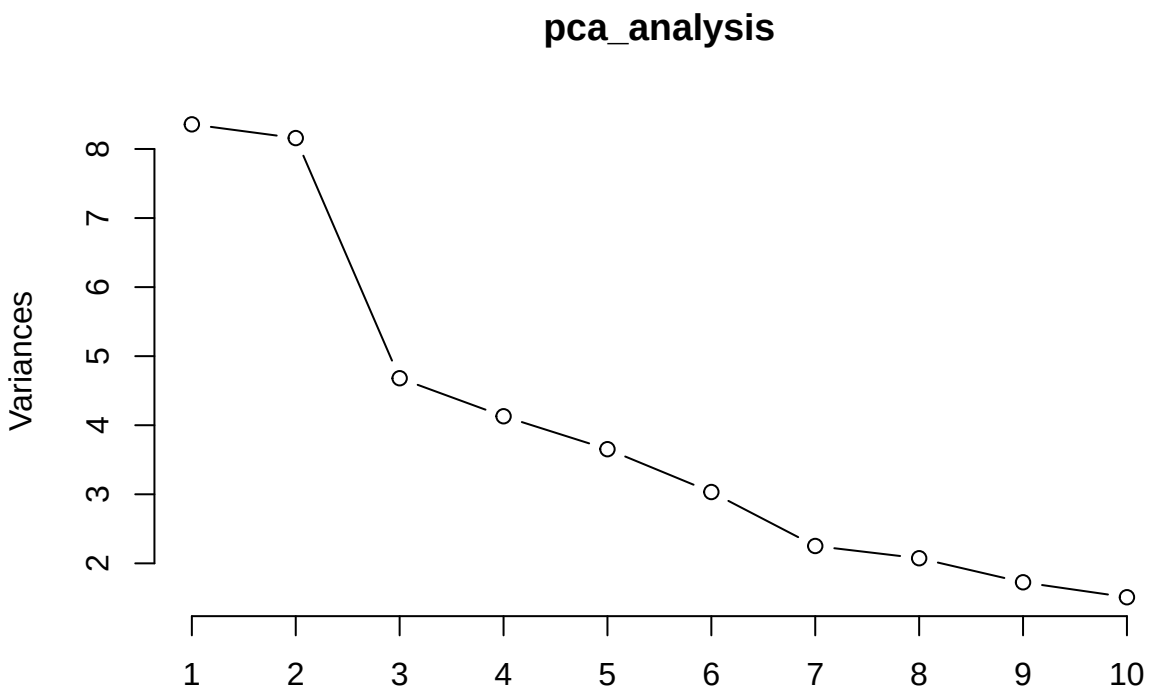
```
pca_analysis <- prcomp(as1_train[1:54], scale.=TRUE)

#!summary(pca_analysis)

sum((pca_analysis$sdev)^2)
```

```
## [1] 54
```

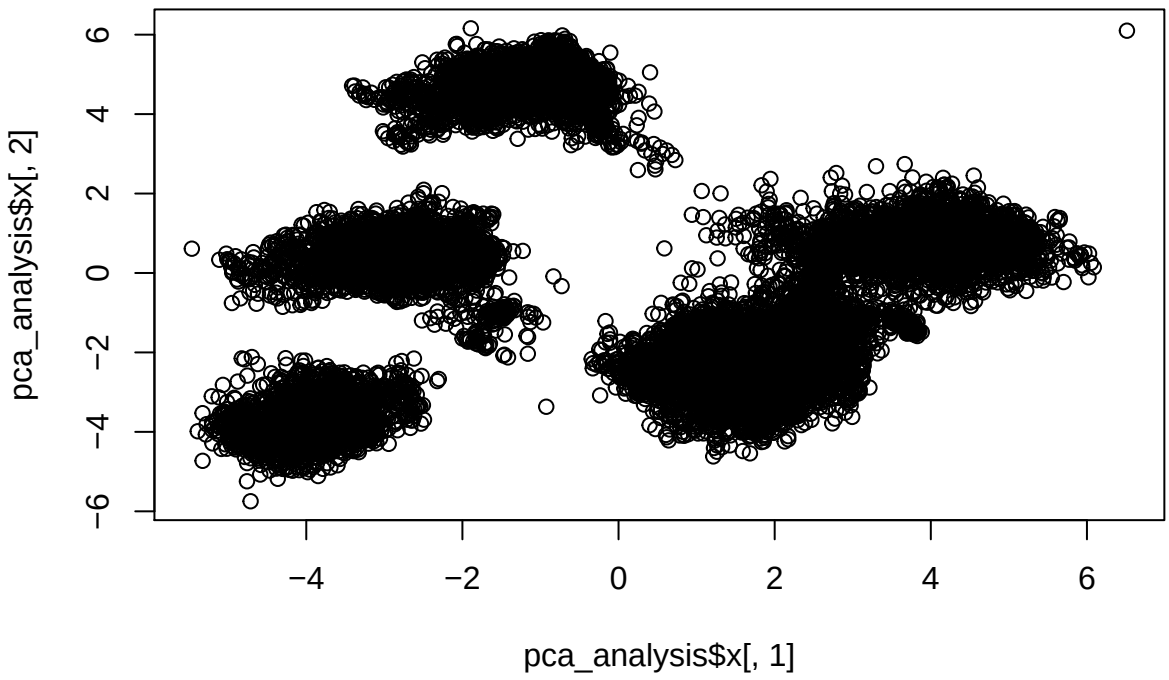
```
screeplot(pca_analysis, type="lines")
```




```
(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 takin
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.

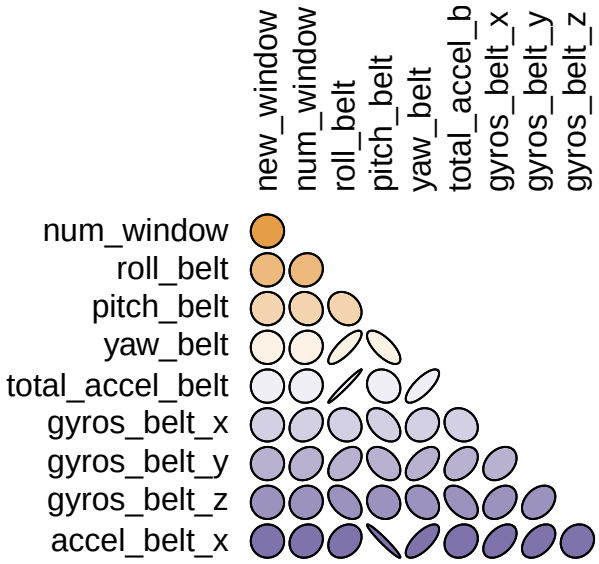
```
# corrgram(as1_train, order= NULL, lower.panel=panel.shade,
#   upper.panel =  NULL, text.panel=panel.txt,
#   main="Groupware  Human Activity Recognition Data in PC2/PC1 Order")
# another colored corrgram

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/multivariat


# 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 |      5535 |        21 |         8 |        11 |         5 |      5580 |
##                 |     0.992 |     0.004 |     0.001 |     0.002 |     0.001 |     0.284 |
##                 |     0.992 |     0.006 |     0.002 |     0.003 |     0.001 |           |
##                 |     0.282 |     0.001 |     0.000 |     0.001 |     0.000 |           |
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|
##               B |        30 |      3699 |        47 |        14 |         7 |      3797 |
##                 |     0.008 |     0.974 |     0.012 |     0.004 |     0.002 |     0.194 |
##                 |     0.005 |     0.982 |     0.013 |     0.004 |     0.002 |           |
##                 |     0.002 |     0.189 |     0.002 |     0.001 |     0.000 |           |
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|
##               C |         3 |        22 |      3368 |        24 |         5 |      3422 |
##                 |     0.001 |     0.006 |     0.984 |     0.007 |     0.001 |     0.174 |
##                 |     0.001 |     0.006 |     0.962 |     0.008 |     0.001 |           |
##                 |     0.000 |     0.001 |     0.172 |     0.001 |     0.000 |           |
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|
##               D |         9 |         9 |        63 |      3126 |         9 |      3216 |
##                 |     0.003 |     0.003 |     0.020 |     0.972 |     0.003 |     0.164 |
##                 |     0.002 |     0.002 |     0.018 |     0.980 |     0.003 |           |
##                 |     0.000 |     0.000 |     0.003 |     0.159 |     0.000 |           |
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|
##               E |         3 |        14 |        14 |        14 |      3562 |      3607 |
##                 |     0.001 |     0.004 |     0.004 |     0.004 |     0.988 |     0.184 |
##                 |     0.001 |     0.004 |     0.004 |     0.004 |     0.993 |           |
##                 |     0.000 |     0.001 |     0.001 |     0.001 |     0.182 |           |
## ----------------|-----------|-----------|-----------|-----------|-----------|-----------|
##    Column Total |      5580 |      3765 |      3500 |      3189 |      3588 |     19622 |
##                 |     0.284 |     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.