HW9_social_network_Yuefei_Chen

Yuefei Chen

2024-04-25

Model development

Running the following code, we build a linear discriminant analysis model to classify social media data. Its independent variables "Instagram_value", "Linkedin_value", "Snapchat_value", "Twitter_value", "Whatsapp_Wechat_value", "Youtube_value", "OTT_Netflix_Hulu_Prime_video_value", "Reddit_value", "job_interview_calls", "networking_done_with_coffee_chats", "learning_done_in_terms_of_items_created". The dependent variable is "Tired_waking_up_in_morning".

```
library(MASS)
library(ggplot2)
library(memisc)
```

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

```
##
## Attaching package: 'memisc'
```

```
## The following object is masked from 'package:ggplot2':
##
##     syms
```

```
## The following objects are masked from 'package:stats':
##
##     contr.sum, contr.treatment, contrasts
```

```
## The following object is masked from 'package:base':
##
##     as.array
```

```
library(ROCR)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:memisc':
##
##     collect, recode, rename, syms
```

```
## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(klaR)
library(readr)
APP_data <- read_csv("Dataset/Social Media_cleaned.csv")
```

```
## New names:
## * `Hours_spent` -> `Hours_spent...3`
## * `Hours_spent` -> `Hours_spent...6`
## * `Hours_spent` -> `Hours_spent...9`
## * `Hours_spent` -> `Hours_spent...15`
## * `Hours_spent` -> `Hours_spent...18`
## * `Hours_spent` -> `Hours_spent...21`
## * `Hours_spent` -> `Hours_spent...24`

## Rows: 23 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (15): ID, Instagram, Linkedin, Snapchat, Twitter, Whatsapp_Wechat, Yout...
## dbl  (12): Instagram_value, Linkedin_value, Snapchat_value, Twitter_value, W...
## time  (6): Hours_spent...3, Hours_spent...6, Hours_spent...9, Hours spent, H...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
APP_data <- APP_data[c(1:22), c(1:2, 4:5, 7:8, 10:11, 13:14, 16:17, 19:20, 22:23, 25:33)]
str(APP_data)
```

```
## tibble [22 x 25] (S3: tbl_df/tbl/data.frame)
##  $ ID                                    : chr [1:22] "masinl" "peace" "Patty" "Bunny" ...
##  $ Instagram                             : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Instagram_value                       : num [1:22] 3.5 7.73 3.77 5.38 0 2.33 5.37 7 8.65 0.17
##  $ Linkedin                              : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Linkedin_value                        : num [1:22] 4 5.2 7 5.32 0.58 7 4 4 10 0 ...
##  $ Snapchat                              : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Snapchat_value                        : num [1:22] 1 3.68 0.53 1.3 0 0.47 0 3 3.83 0 ...
##  $ Twitter                               : chr [1:22] "Yes" "No" "No" "No" ...
##  $ Twitter_value                         : num [1:22] 5 0 0 0 0.67 0 0 0 0 0 ...
##  $ Whatsapp_Wechat                       : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Whatsapp_Wechat_value                 : num [1:22] 1 4.18 9.83 5.3 3 12 6 10 6.15 1 ...
##  $ Youtube                               : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Youtube_value                         : num [1:22] 2.5 4.25 1.85 2 3.5 7 3 2 4 3 ...
```

```
##  $ OTT_Netflix_Hulu_Prime video              : chr [1:22] "Yes" "No" "Yes" "Yes" ...
##  $ OTT_Netflix_Hulu_Prime_video_value        : num [1:22] 14.5 0 2 2 2 3 0 3 3 0 ...
##  $ Reddit                                    : chr [1:22] "Yes" "No" "No" "No" ...
##  $ Reddit_value                              : num [1:22] 2.5 0 0 0 1 0 0 0 0 0 ...
##  $ Application_type_Social media_OTT_Learning: chr [1:22] "OTT" "Social Media" "Social Media" "Social
##  $ job_interview_calls                       : num [1:22] 0 0 0 2 0 0 0 0 1 0 ...
##  $ networking_done_with_coffee_chats         : num [1:22] 0 1 0 0 2 0 2 0 0 0 ...
##  $ learning_done_in_terms_of_items_created   : num [1:22] 3 3 4 4 4 4 3 2 6 2 ...
##  $ Mood_Productivity                         : chr [1:22] "Yes" "Yes" "Yes" "Yes" ...
##  $ Tired_waking_up_in_morning                : chr [1:22] "No" "No" "No" "No" ...
##  $ Trouble_falling_asleep                    : chr [1:22] "No" "Yes" "No" "No" ...
##  $ felt_the_entire_week                      : num [1:22] 3 3 4 4 3 5 4 4 3 2 ...
```

```
APP_data$Tired_waking_up_in_morning <- as.factor(APP_data$Tired_waking_up_in_morning)
r <- lda(formula = Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twi
head(r$class)
```

```
## NULL
```

```
summary(r)
```

```
##         Length Class  Mode
## prior    2     -none- numeric
## counts   2     -none- numeric
## means   20     -none- numeric
## scaling 10     -none- numeric
## lev      2     -none- character
## svd      1     -none- numeric
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  0     -none- list
```

Model Acceptance

In this model, we can see that the first linear discriminant explains all the between-group variance in the house data. Therefore, the model can be used to analyze the house data.

```
r$svd
```

```
## [1] 4.219726
```

```
(prop = r$svd^2/sum(r$svd^2))
```

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

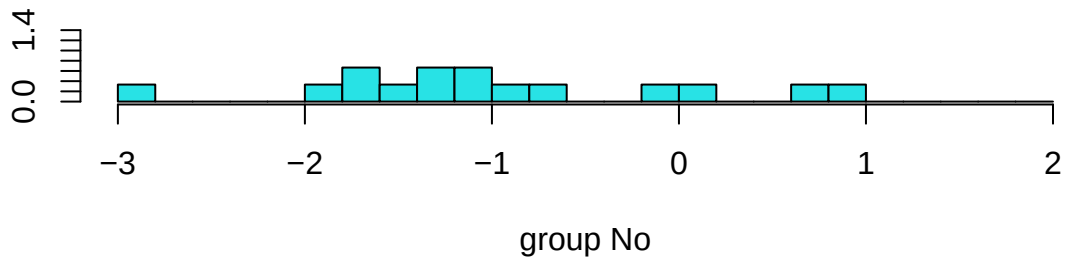
Residual Analysis

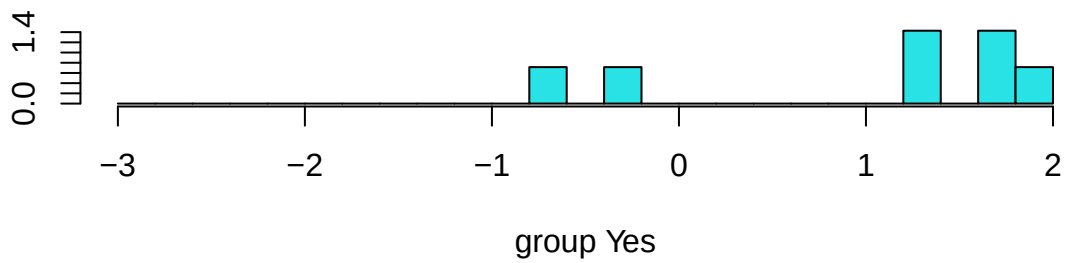
Since this model is a classification model, we focus on the posterior value of the model. The following code is to train the new model r3 and the model is used to test the model and display the predicted result and posterior probability. The plots of r1 and r3 shows how the model distinguishes between different furniture categories on training data

```
r2 <- lda(formula = Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twi
head(r2$posterior, 3)
```

```
##          No        Yes
## 1 0.9953703 0.00462970
## 2 0.9761574 0.02384258
## 3 0.1196170 0.88038297
```

```
plot(r)
```

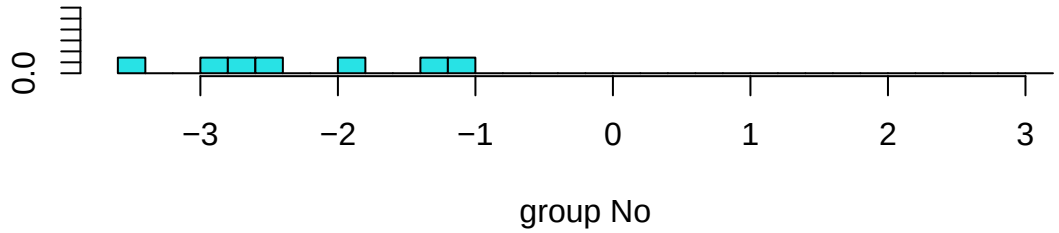




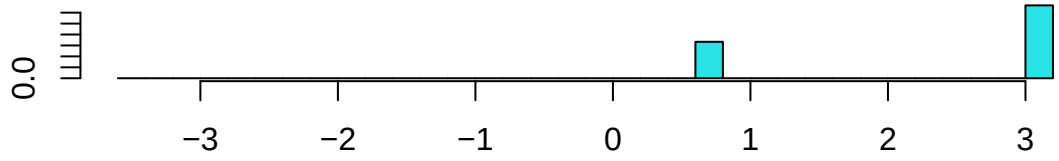

```
train <- sample(22, 10)
r3 <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twitter_valu
          APP_data,
          prior = c(1,1)/2,
          subset = train)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
plot(r3)
```



group No



group Yes

```
plda = predict(object = r3, # predictions
               newdata = APP_data[-train, ])
head(plda$class)
```

```
## [1] Yes No  No  Yes No  Yes
## Levels: No Yes
```

```
head(plda$posterior, 6) # posterior prob.
```

```
##             No         Yes
## 1 2.406362e-87 1.000000000
## 2 9.989909e-01 0.001009120
## 3 9.658094e-01 0.034190587
## 4 6.965192e-40 1.000000000
## 5 9.935560e-01 0.006444047
## 6 4.116359e-03 0.995883641
```

```
head(plda$x, 3)
```

```
##          LD1
## 1 43.2075207
## 2 -1.4942887
## 3 -0.7237872
```

Prediction

The data will be predicted in the model and the predicted first linear discriminant scores of the are as follows.

```
r <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twitter_value
         APP_data,
         prior = c(1,1)/2,)
prop.lda = r$svd^2/sum(r$svd^2)
plda <- predict(object = r,
                newdata = APP_data)
dataset = data.frame(furniture = APP_data[,"Tired_waking_up_in_morning"],lda = plda$x)
dataset$LD1
```

```
##  [1] -1.64940523 -1.50160901  0.43417209  0.07298424  1.88111904 -3.14085528
##  [7]  1.36594301 -0.18538993 -1.02890956  0.71923625 -2.10060140 -1.60142155
## [13]  1.41842173 -0.75351014 -1.35531594 -0.66500724 -0.69634235  1.42556247
## [19] -0.12856207 -1.01810093  1.70927156 -1.14444620
```

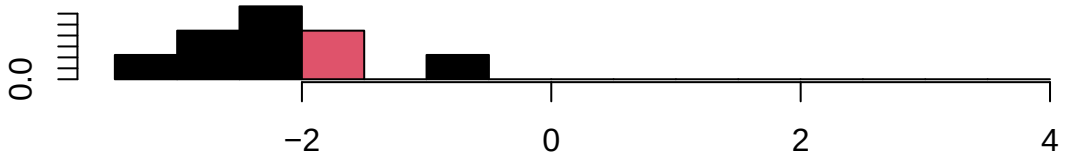
Model Accuracy

To observe the performance of the model, the test set is used to approximate accuracy.

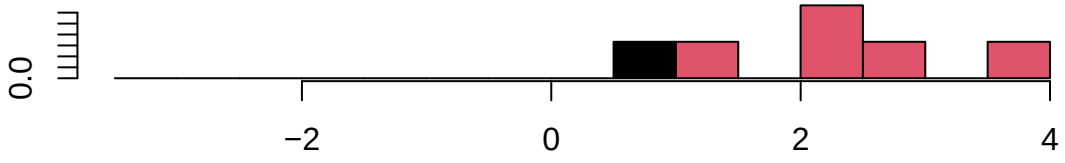
```
set.seed(101)
sample_n(APP_data,10)
```

```
## # A tibble: 10 x 25
##    ID      Instagram Instagram_value Linkedin Linkedin_value Snapchat
##    <chr>   <chr>               <dbl> <chr>             <dbl> <chr>
##  1 yh2020  Yes                  8.65 Yes               10    Yes
##  2 hahah   Yes                  6    Yes                3    Yes
##  3 sss32   Yes                  9.8  Yes                0.8  No
##  4 Patty   Yes                  3.77 Yes                7    Yes
##  5 2134    Yes                  5.67 Yes                3.92 No
##  6 azhena  Yes                  8    Yes                2    No
##  7 vp1234  Yes                  7    Yes                5    yes
##  8 MVA37@S Yes                  6.8  Yes                1.92 Yes
##  9 AKIRA   Yes                  4.65 Yes                3.75 Yes
## 10 peace   Yes                  7.73 Yes                5.2  Yes
## # i 19 more variables: Snapchat_value <dbl>, Twitter <chr>,
## #   Twitter_value <dbl>, Whatsapp_Wechat <chr>, Whatsapp_Wechat_value <dbl>,
## #   Youtube <chr>, Youtube_value <dbl>, `OTT_Netflix_Hulu_Prime video` <chr>,
## #   OTT_Netflix_Hulu_Prime_video_value <dbl>, Reddit <chr>, Reddit_value <dbl>,
## #   `Application_type_Social media_OTT_Learning` <chr>,
## #   job_interview_calls <dbl>, networking_done_with_coffee_chats <dbl>,
## #   learning_done_in_terms_of_items_created <dbl>, Mood_Productivity <chr>, ...
```

```
training_sample <- sample(c(TRUE, FALSE), nrow(APP_data), replace = T, prob = c(0.75,0.25))
train <- APP_data[training_sample, ]
test <- APP_data[!training_sample, ]
lda.waking <- lda(Tired_waking_up_in_morning ~ Instagram_value + Linkedin_value + Snapchat_value + Twit
plot(lda.waking, col = as.integer(train$Tired_waking_up_in_morning))
```

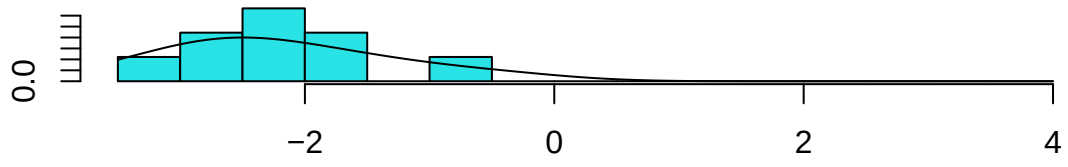


group No

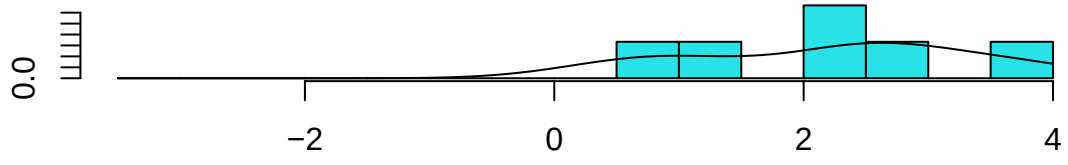


group Yes

```
# Sometime bell curves are better
plot(lda.waking, dimen = 1, type = "b")
```



group No



group Yes

```
lda.train <- predict(lda.waking)
train$lda <- lda.train$class
table(train$lda,train$Tired_waking_up_in_morning)
```

```
##
##       No Yes
##   No   9   0
##   Yes  0   6
```

```
# running accuracy on the training set shows how good the model is. It is not an indication of "true" a
lda.test <- predict(lda.waking,test)
test$lda <- lda.test$class
table(test$lda,test$Tired_waking_up_in_morning)
```

```
##
##       No Yes
##   No   1   0
##   Yes  5   1
```