Dimensonality_and_feature_selection

GRIFFIN BURET

2022-08-04

##RESEARCH QUESTION##

Carrefour Kenya and are currently undertaking a project that will inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).This project is aimed at doing analysis on the dataset provided by carrefour and create insights on how to achieve highest sales.

##METRIC FOR SUCCESS##

Be able to detect and do away with anomalies in our dataset

##THE CONTEXT##

Carre Four is an International chain of retail supemarkets in the world, It was set up in Kenya in the year 2016 and has been performing well over the years.Carrefour ensures customer satisfaction and everyday convenience while offering unbeatable value for money with a vast array of more than 100,000 products, shoppers can purchase items for their every need, whether home electronics or fresh fruits from around the world, to locally produced items. This project is aimed at creating insights from existing and current trends to develop marketing strategies that will enable the marketing team achieve higher sales.

##EXPERIMENTAL DESIGN##

1. Loading libraries
2. Load data
3. Data cleaning
4. PCA 5.Feature selection
5. Conclusion
6. Recommendation # importing libraries

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(superml)

## Loading required package: R6
```

```
#install.packages("devtools", type = "win.binary")
remotes::install_github('vqv/ggbiplot')

## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e
880) has not changed since last install.
##   Use `force = TRUE` to force installation

library(ggbiplot)

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid

#suppressWarnings(
        #suppressMessages(if
                         #(!require(corrplot, quietly=TRUE))
                #install.packages("corrplot")))
library(corrplot)

## corrplot 0.92 loaded
```

Loading dataset

```
#reading the data set
sales<- read.csv("C:/Users/Admin/Downloads/Supermarket_Dataset_1 - Sale
s Data.csv")
```

Previewing the data

```
#previewing the head
head(sales)

##    Invoice.ID Branch Customer.type Gender           Product.line Uni
t.price
## 1 750-67-8428      A        Member Female      Health and beauty
  74.69
## 2 226-31-3081      C        Normal Female Electronic accessories
  15.28
## 3 631-41-3108      A        Normal   Male     Home and lifestyle
  46.33
## 4 123-19-1176      A        Member   Male      Health and beauty
  58.22
## 5 373-73-7910      A        Normal   Male      Sports and travel
  86.31
## 6 699-14-3026      C        Normal   Male Electronic accessories
  85.39
##   Quantity     Tax      Date  Time     Payment   cogs gross.margin.p
ercentage
## 1        7 26.1415  1/5/2019 13:08     Ewallet 522.83
 4.761905
```

```
## 2        5  3.8200  3/8/2019 10:29        Cash  76.40
 4.761905
## 3        7 16.2155  3/3/2019 13:23 Credit card 324.31
 4.761905
## 4        8 23.2880 1/27/2019 20:33     Ewallet 465.76
 4.761905
## 5        7 30.2085  2/8/2019 10:37     Ewallet 604.17
 4.761905
## 6        7 29.8865 3/25/2019 18:30     Ewallet 597.73
 4.761905
##   gross.income Rating    Total
## 1      26.1415    9.1 548.9715
## 2       3.8200    9.6  80.2200
## 3      16.2155    7.4 340.5255
## 4      23.2880    8.4 489.0480
## 5      30.2085    5.3 634.3785
## 6      29.8865    4.1 627.6165
```

```
#previewing the tail
tail(sales)
```

```
##       Invoice.ID Branch Customer.type Gender           Product.line
Unit.price
## 995  652-49-6720      C        Member Female Electronic accessories
     60.95
## 996  233-67-5758      C        Normal   Male      Health and beauty
     40.35
## 997  303-96-2227      B        Normal Female     Home and lifestyle
     97.38
## 998  727-02-1313      A        Member   Male     Food and beverages
     31.84
## 999  347-56-2442      A        Normal   Male     Home and lifestyle
     65.82
## 1000 849-09-3807      A        Member Female    Fashion accessories
     88.34
##      Quantity     Tax      Date  Time Payment   cogs gross.margin.pe
rcentage
## 995         1  3.0475 2/18/2019 11:40 Ewallet  60.95
4.761905
## 996         1  2.0175 1/29/2019 13:46 Ewallet  40.35
4.761905
## 997        10 48.6900  3/2/2019 17:16 Ewallet 973.80
4.761905
## 998         1  1.5920  2/9/2019 13:22    Cash  31.84
4.761905
## 999         1  3.2910 2/22/2019 15:33    Cash  65.82
4.761905
## 1000        7 30.9190 2/18/2019 13:28    Cash 618.38
4.761905
##      gross.income Rating     Total
```

```
## 995        3.0475    5.9   63.9975
## 996        2.0175    6.2   42.3675
## 997       48.6900    4.4 1022.4900
## 998        1.5920    7.7   33.4320
## 999        3.2910    4.1   69.1110
## 1000      30.9190    6.6  649.2990

#checking the data structure
str(sales)

## 'data.frame':    1000 obs. of  16 variables:
##  $ Invoice.ID             : chr  "750-67-8428" "226-31-3081" "631-41
-3108" "123-19-1176" ...
##  $ Branch                 : chr  "A" "C" "A" "A" ...
##  $ Customer.type          : chr  "Member" "Normal" "Normal" "Member"
 ...
##  $ Gender                 : chr  "Female" "Female" "Male" "Male" ...
##  $ Product.line           : chr  "Health and beauty" "Electronic acc
essories" "Home and lifestyle" "Health and beauty" ...
##  $ Unit.price             : num  74.7 15.3 46.3 58.2 86.3 ...
##  $ Quantity               : int  7 5 7 8 7 7 6 10 2 3 ...
##  $ Tax                    : num  26.14 3.82 16.22 23.29 30.21 ...
##  $ Date                   : chr  "1/5/2019" "3/8/2019" "3/3/2019" "1
/27/2019" ...
##  $ Time                   : chr  "13:08" "10:29" "13:23" "20:33" ...
##  $ Payment                : chr  "Ewallet" "Cash" "Credit card" "Ewa
llet" ...
##  $ cogs                   : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ gross.margin.percentage: num  4.76 4.76 4.76 4.76 4.76 ...
##  $ gross.income           : num  26.14 3.82 16.22 23.29 30.21 ...
##  $ Rating                 : num  9.1 9.6 7.4 8.4 5.3 4.1 5.8 8 7.2 5.
9 ...
##  $ Total                  : num  549 80.2 340.5 489 634.4 ...

#checking the unique values in the rows
#sapply(sales,n_distinct)
```

our data has 1000 observations and 16 columns. there are 8 character variables and 8 numeric variables # Data cleaning **Checking missing values**

```
#checking missing values
colSums((is.na(sales)))

##               Invoice.ID                   Branch           Customer.t
ype
##                        0                        0                     
  0
##                   Gender             Product.line              Unit.pr
ice
##                        0                        0                     
  0
```

```
##                Quantity                      Tax                     D
ate
##                       0                        0
  0
##                    Time                  Payment                     c
ogs
##                       0                        0
  0
## gross.margin.percentage             gross.income                   Rat
ing
##                       0                        0
  0
##                   Total
##                       0
```

our dataset has no missing values **checking duplicates**

```
duplicates <- sales[duplicated(sales)]
duplicates

## data frame with 0 columns and 1000 rows
```

our data has no duplicates

```
#dropping columns we wont nedd
#we drop the id column , date column,time and gross.margin column  sinc
e we wont need
sales_df<- sales[,-c(1,9,10,13)]
head(sales_df)

##   Branch Customer.type Gender           Product.line Unit.price Quan
tity
## 1      A        Member Female      Health and beauty      74.69
   7
## 2      C        Normal Female Electronic accessories      15.28
   5
## 3      A        Normal   Male     Home and lifestyle      46.33
   7
## 4      A        Member   Male      Health and beauty      58.22
   8
## 5      A        Normal   Male      Sports and travel      86.31
   7
## 6      C        Normal   Male Electronic accessories      85.39
   7
##       Tax     Payment   cogs gross.income Rating    Total
## 1 26.1415     Ewallet 522.83      26.1415    9.1 548.9715
## 2  3.8200        Cash  76.40       3.8200    9.6  80.2200
## 3 16.2155 Credit card 324.31      16.2155    7.4 340.5255
## 4 23.2880     Ewallet 465.76      23.2880    8.4 489.0480
## 5 30.2085     Ewallet 604.17      30.2085    5.3 634.3785
## 6 29.8865     Ewallet 597.73      29.8865    4.1 627.6165
```

Data Processing

converting categorical data to numeric

```
#label encoding our data set
label <- LabelEncoder$new()
print(label$fit(sales_df$Customer.type))

## [1] TRUE

print(label$fit(sales_df$Gender))

## [1] TRUE

print(label$fit(sales_df$Product.line))

## [1] TRUE

print(label$fit(sales_df$Payment))

## [1] TRUE

sales_df$Branch <- label$fit_transform(sales_df$Branch)
sales_df$Customer.type <- label$fit_transform(sales_df$Customer.type)
sales_df$Gender <- label$fit_transform(sales_df$Gender)
sales_df$Product.line <- label$fit_transform(sales_df$Product.line)
sales_df$Payment <- label$fit_transform(sales_df$Payment)
head(sales_df)

##   Branch Customer.type Gender Product.line Unit.price Quantity     T
ax Payment
## 1      0             0      0            0      74.69        7 26.14
15       0
## 2      1             1      0            1      15.28        5  3.82
00       1
## 3      0             1      1            2      46.33        7 16.21
55       2
## 4      0             0      1            0      58.22        8 23.28
80       0
## 5      0             1      1            3      86.31        7 30.20
85       0
## 6      1             1      1            1      85.39        7 29.88
65       0
##     cogs gross.income Rating    Total
## 1 522.83      26.1415    9.1 548.9715
## 2  76.40       3.8200    9.6  80.2200
## 3 324.31      16.2155    7.4 340.5255
## 4 465.76      23.2880    8.4 489.0480
## 5 604.17      30.2085    5.3 634.3785
## 6 597.73      29.8865    4.1 627.6165
```

scaling data

```r
#we encoded our data because PCA only works with numeric data and since
 it is sensitive to scale of measurement we need to scale our data
sales_num<- sales_df[,c(5,7:12)]
head(sales_num)

##   Unit.price     Tax Payment   cogs gross.income Rating    Total
## 1      74.69 26.1415       0 522.83      26.1415    9.1 548.9715
## 2      15.28  3.8200       1  76.40       3.8200    9.6  80.2200
## 3      46.33 16.2155       2 324.31      16.2155    7.4 340.5255
## 4      58.22 23.2880       0 465.76      23.2880    8.4 489.0480
## 5      86.31 30.2085       0 604.17      30.2085    5.3 634.3785
## 6      85.39 29.8865       0 597.73      29.8865    4.1 627.6165

#checking the stats of our numerical data to check if they have same me
an and variance
stats<- data.frame(
  sd=apply(sales_num,2,sd),
  mean = apply(sales_num,2,mean)

)
stats

##                       sd      mean
## Unit.price    26.4946283  55.67213
## Tax           11.7088255  15.37937
## Payment        0.8096292   0.96600
## cogs         234.1765096 307.58738
## gross.income  11.7088255  15.37937
## Rating         1.7185803   6.97270
## Total        245.8853351 322.96675

#the numerical dataset has different means and variance thus the need t
o scale
sales_scale<- scale(sales_num)
head(sales_scale)

##       Unit.price         Tax     Payment        cogs gross.income
  Rating
## [1,]  0.71780097  0.91914693 -1.19313873  0.91914693   0.91914693  1.
2378240
## [2,] -1.52454035 -0.98723557  0.04199453 -0.98723557  -0.98723557  1.
5287619
## [3,] -0.35260468  0.07141032  1.27712779  0.07141032   0.07141032  0.
2486355
## [4,]  0.09616553  0.67544187 -1.19313873  0.67544187   0.67544187  0.
8305111
## [5,]  1.15638044  1.26649176 -1.19313873  1.26649176   1.26649176 -0.
9733034
## [6,]  1.12165642  1.23899114 -1.19313873  1.23899114   1.23899114 -1.
6715541
##            Total
```

```
## [1,]  0.91914693
## [2,] -0.98723557
## [3,]  0.07141032
## [4,]  0.67544187
## [5,]  1.26649176
## [6,]  1.23899114
```

```
#combining the numerical data with the categorical
sales_new<- cbind(sales_df,sales_scale)
sales_data<- sales_new[,-c(5,7:12)]
head(sales_data)
```

```
##   Branch Customer.type Gender Product.line Quantity  Unit.price
    Tax
## 1      0             0      0            0        7  0.71780097  0.9
1914693
## 2      1             1      0            1        5 -1.52454035 -0.9
8723557
## 3      0             1      1            2        7 -0.35260468  0.0
7141032
## 4      0             0      1            0        8  0.09616553  0.6
7544187
## 5      0             1      1            3        7  1.15638044  1.2
6649176
## 6      1             1      1            1        7  1.12165642  1.2
3899114
##       Payment        cogs gross.income     Rating       Total
## 1 -1.19313873  0.91914693   0.91914693  1.2378240  0.91914693
## 2  0.04199453 -0.98723557  -0.98723557  1.5287619 -0.98723557
## 3  1.27712779  0.07141032   0.07141032  0.2486355  0.07141032
## 4 -1.19313873  0.67544187   0.67544187  0.8305111  0.67544187
## 5 -1.19313873  1.26649176   1.26649176 -0.9733034  1.26649176
## 6 -1.19313873  1.23899114   1.23899114 -1.6715541  1.23899114
```

implementing the solution

Dimensionality Reduction usinng PCA

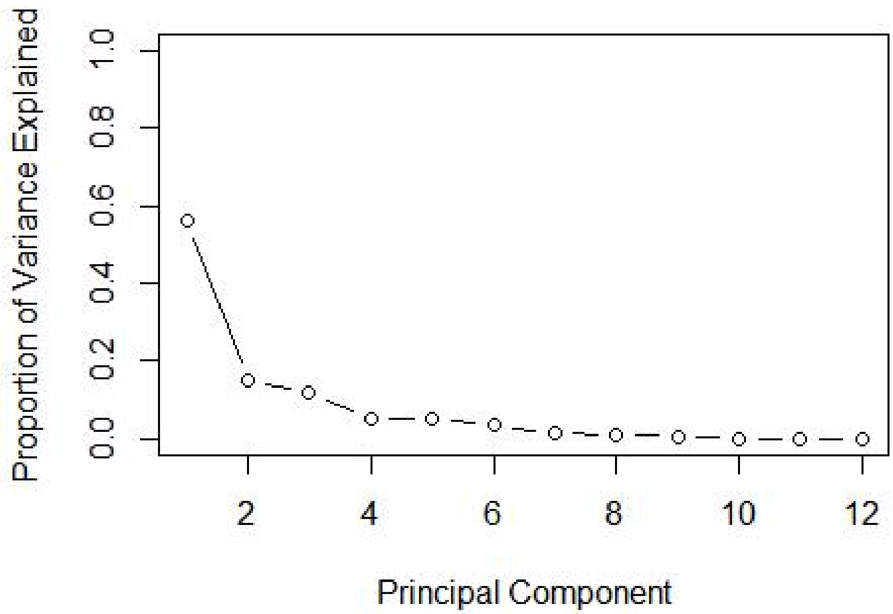
```
#fitting the model
sales_pca <- prcomp(sales_data,scale=FALSE,center=TRUE)
summary(sales_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
    PC7
## Standard deviation     3.3229 1.7146 1.5308 1.00619 0.99747 0.81592
0.50549
## Proportion of Variance 0.5623 0.1497 0.1193 0.05156 0.05067 0.03391
0.01301
## Cumulative Proportion  0.5623 0.7121 0.8314 0.88297 0.93364 0.96754
```

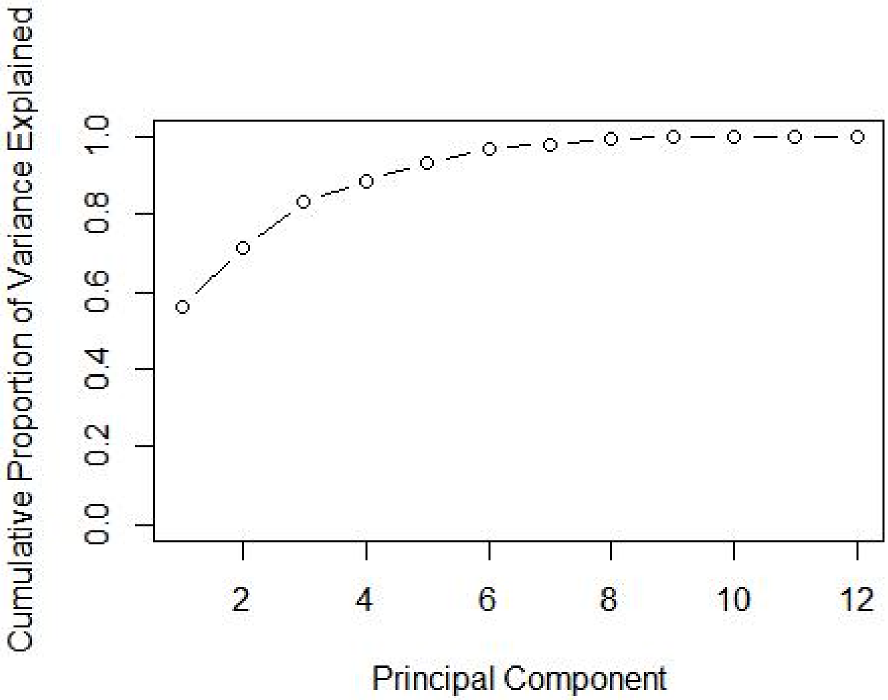
```
0.98056
##                               PC8     PC9      PC10      PC11      PC12
## Standard deviation     0.4895 0.37696 2.049e-16 1.363e-16 1.166e-16
## Proportion of Variance 0.0122 0.00724 0.000e+00 0.000e+00 0.000e+00
## Cumulative Proportion  0.9928 1.00000 1.000e+00 1.000e+00 1.000e+00
```

our data has 12 PCs and the the first,second and the third explain 56%,14% and 12% variance respectively

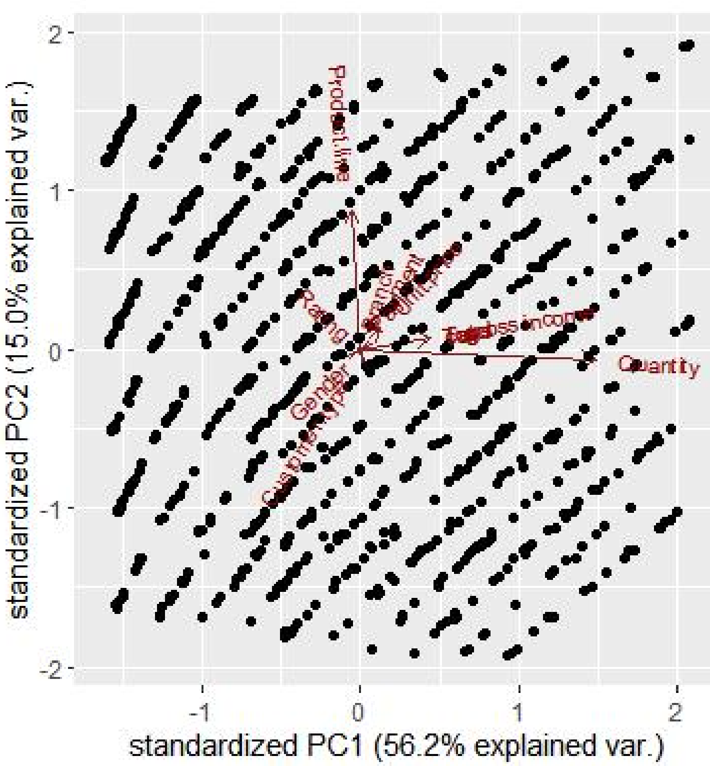
```
#getting the structure of the PCA  output to see the sdev ,rotation and
 other output
str(sales_pca)

## List of 5
##  $ sdev    : num [1:12] 3.323 1.715 1.531 1.006 0.997 ...
##  $ rotation: num [1:12, 1:12] 0.00151 -0.00289 -0.01073 -0.03623 0.8
5469 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12] "Branch" "Customer.type" "Gender" "Product.lin
e" ...
##   .. ..$ : chr [1:12] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:12] 0.992 0.499 0.499 2.574 5.51 ...
##   ..- attr(*, "names")= chr [1:12] "Branch" "Customer.type" "Gender"
 "Product.line" ...
##  $ scale   : logi FALSE
##  $ x       : num [1:1000, 1:12] 2.35 -1.5 1.34 2.91 2.63 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:12] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"

#ploting a scree plot to see the variation of each PC
#getting the variance
pr<- sales_pca$sdev^2
#getting propotion
pve<- pr/sum(pr)
#ploting scree plot
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "b")
```




```
# Plot cumulative proportion of variance explained
plot(cumsum(pve), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     ylim = c(0, 1), type = "b")
```

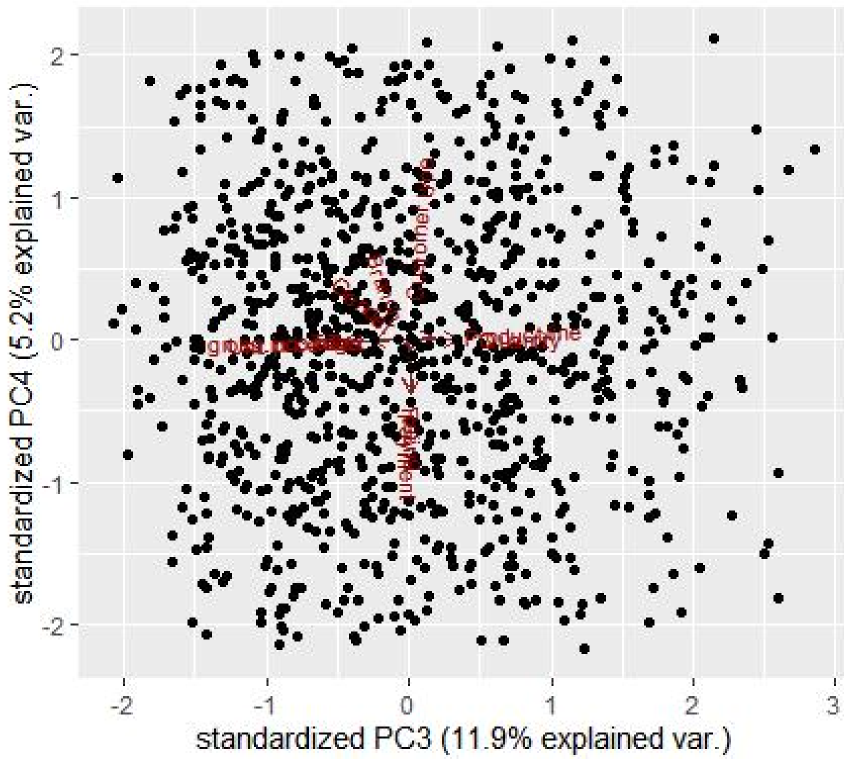



```
#ploting biplot using the first 2 PCs
ggbiplot(sales_pca)
```



from the biplot we can see that the quantity and the product line has the longest vectors thus they contribute the most variability the quantity contributes more to PC1 and the product line contribute more to PC2 the gross income and the quantity are closely corelated as the angle between the vectors are small

```
#ploting biplot using the third and fourth  PCs
ggbiplot(sales_pca,choices = c(3,4))
```



the length of the vectors are short since most variability have been accounted by the first two components **feature selection**

```
# calculate correlation matrix
correlationMatrix <- cor(sales_data)
# summarize the correlation matrix
print(correlationMatrix)

##                     Branch Customer.type       Gender Product.line
   Quantity
## Branch         1.000000000  -0.004899261 -0.012218875   0.01257525
0.002120920
## Customer.type -0.004899261   1.000000000  0.039996160  -0.02510945 -
0.016762706
## Gender        -0.012218875   0.039996160  1.000000000  -0.06612647 -
0.074258307
## Product.line   0.012575246  -0.025109450 -0.066126475   1.00000000 -
0.062514713
## Quantity       0.002120920  -0.016762706 -0.074258307  -0.06251471
1.000000000
## Unit.price     0.013763477  -0.020237875  0.015444630   0.03842765
```

```
0.010777564
## Tax           0.012811933  -0.019670283 -0.049450989  -0.01854396 
0.705510186
## Payment       0.026725563  -0.069286242 -0.049514182   0.01051098 
0.007333388
## cogs          0.012811933  -0.019670283 -0.049450989  -0.01854396 
0.705510186
## gross.income  0.012811933  -0.019670283 -0.049450989  -0.01854396 
0.705510186
## Rating       -0.049585348   0.018888672  0.004800208   0.02339096 -
0.015814905
## Total         0.012811933  -0.019670283 -0.049450989  -0.01854396 
0.705510186
##                Unit.price          Tax      Payment         cogs gr
oss.income
## Branch        0.013763477  0.012811933  0.026725563  0.012811933  0.
012811933
## Customer.type -0.020237875 -0.019670283 -0.069286242 -0.019670283 -0.
019670283
## Gender        0.015444630 -0.049450989 -0.049514182 -0.049450989 -0.
049450989
## Product.line  0.038427649 -0.018543956  0.010510982 -0.018543956 -0.
018543956
## Quantity      0.010777564  0.705510186  0.007333388  0.705510186  0.
705510186
## Unit.price    1.000000000  0.633962089 -0.019637884  0.633962089  0.
633962089
## Tax           0.633962089  1.000000000  0.008823723  1.000000000  1.
000000000
## Payment      -0.019637884  0.008823723  1.000000000  0.008823723  0.
008823723
## cogs          0.633962089  1.000000000  0.008823723  1.000000000  1.
000000000
## gross.income  0.633962089  1.000000000  0.008823723  1.000000000  1.
000000000
## Rating       -0.008777507 -0.036441705  0.013001094 -0.036441705 -0.
036441705
## Total         0.633962089  1.000000000  0.008823723  1.000000000  1.
000000000
##                    Rating        Total
## Branch       -0.049585348  0.012811933
## Customer.type  0.018888672 -0.019670283
## Gender        0.004800208 -0.049450989
## Product.line  0.023390962 -0.018543956
## Quantity     -0.015814905  0.705510186
## Unit.price   -0.008777507  0.633962089
## Tax          -0.036441705  1.000000000
## Payment       0.013001094  0.008823723
## cogs         -0.036441705  1.000000000
## gross.income -0.036441705  1.000000000
```

```
## Rating         1.000000000 -0.036441705
## Total         -0.036441705  1.000000000
```

```
# find attributes that are highly corrected (ideally >0.75)
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.75)
# print indexes of highly correlated attributes
print(highlyCorrelated)
```

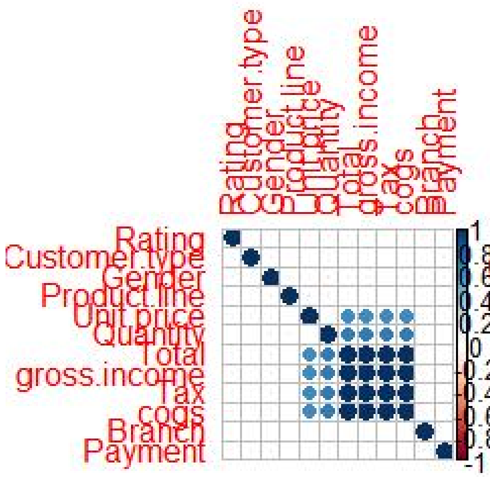
```
## [1]  7  9 10
```

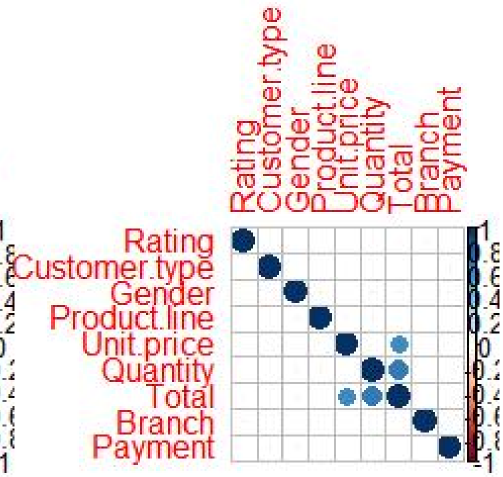
we see that the highly correlated attributes are tax,cogs,gross.income and thus we remove them

```
#we now remove the highly correlleted variables
sales_data1<- sales_data[-(highlyCorrelated)]
head(sales_data1)
```

```
##   Branch Customer.type Gender Product.line Quantity  Unit.price
Payment
## 1      0             0      0            0        7  0.71780097 -1.1
9313873
## 2      1             1      0            1        5 -1.52454035  0.0
4199453
## 3      0             1      1            2        7 -0.35260468  1.2
7712779
## 4      0             0      1            0        8  0.09616553 -1.1
9313873
## 5      0             1      1            3        7  1.15638044 -1.1
9313873
## 6      1             1      1            1        7  1.12165642 -1.1
9313873
##       Rating       Total
## 1  1.2378240  0.91914693
## 2  1.5287619 -0.98723557
## 3  0.2486355  0.07141032
## 4  0.8305111  0.67544187
## 5 -0.9733034  1.26649176
## 6 -1.6715541  1.23899114
```

```
#making graphical presantation before and after removing the highly cor
related features
par(mfrow = c(1, 2))
corrplot(correlationMatrix, order = "hclust")
corrplot(cor(sales_data1), order = "hclust")
```





#
Recomendations the are some variables that are redundant thus need to do dimensonality reduction and feature selection to identify the important features # Conclusion dimesionality reduction and feature selection helps speed up the training of the model as they remove reduntant features