Part 2: Feature Selection

```
# Import Libraries
library(data.table)
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

library(corrplot)

## corrplot 0.84 loaded

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.
3.0 --

## v tibble  3.1.0     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflict
s() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::lift()      masks caret::lift()
## x purrr::transpose() masks data.table::transpose()
```

```r
# Loading the dataset
df <- read_csv('http://bit.ly/CarreFourDataset')

##
## -- Column specification --------------------------------------------------
------
## cols(
##   `Invoice ID` = col_character(),
##   Branch = col_character(),
##   `Customer type` = col_character(),
##   Gender = col_character(),
##   `Product line` = col_character(),
##   `Unit price` = col_double(),
##   Quantity = col_double(),
##   Tax = col_double(),
##   Date = col_character(),
##   Time = col_time(format = ""),
##   Payment = col_character(),
##   cogs = col_double(),
##   `gross margin percentage` = col_double(),
##   `gross income` = col_double(),
##   Rating = col_double(),
##   Total = col_double()
## )

# Loading the dataset as a dataframe
#df = as.data.frame(data)
# Previewing the first five rows of the dataframe
head(df)

## # A tibble: 6 x 16
##   `Invoice ID` Branch `Customer type` Gender `Product line`         `Unit
price`
##   <chr>        <chr>  <chr>           <chr>  <chr>                    <dbl>
## 1 750-67-8428  A      Member          Female Health and beauty         74.7
## 2 226-31-3081  C      Normal          Female Electronic accessories    15.3
## 3 631-41-3108  A      Normal          Male   Home and lifestyle        46.3
## 4 123-19-1176  A      Member          Male   Health and beauty         58.2
## 5 373-73-7910  A      Normal          Male   Sports and travel         86.3
## 6 699-14-3026  C      Normal          Male   Electronic accessories    85.4
## # ... with 10 more variables: Quantity <dbl>, Tax <dbl>, Date <chr>,
## #   Time <time>, Payment <chr>, cogs <dbl>, gross margin percentage <dbl>,
## #   gross income <dbl>, Rating <dbl>, Total <dbl>
```

```
# displaying all rows from the dataset which don't contain any missing values
na.omit(df)
```

```
## # A tibble: 1,000 x 16
##    `Invoice ID` Branch `Customer type` Gender `Product line`        `Unit
price`
##    <chr>        <chr>  <chr>           <chr>  <chr>
<dbl>
##  1 750-67-8428  A      Member          Female Health and beauty
74.7
##  2 226-31-3081  C      Normal          Female Electronic accessori~
15.3
##  3 631-41-3108  A      Normal          Male   Home and lifestyle
46.3
##  4 123-19-1176  A      Member          Male   Health and beauty
58.2
##  5 373-73-7910  A      Normal          Male   Sports and travel
86.3
##  6 699-14-3026  C      Normal          Male   Electronic accessori~
85.4
##  7 355-53-5943  A      Member          Female Electronic accessori~
68.8
##  8 315-22-5665  C      Normal          Female Home and lifestyle
73.6
##  9 665-32-9167  A      Member          Female Health and beauty
36.3
## 10 692-92-5582  B      Member          Female Food and beverages
54.8
## # ... with 990 more rows, and 10 more variables: Quantity <dbl>, Tax <dbl>
,
## #   Date <chr>, Time <time>, Payment <chr>, cogs <dbl>,
## #   gross margin percentage <dbl>, gross income <dbl>, Rating <dbl>,
## #   Total <dbl>
```

Data cleaning

Columns

```
# Changing column names to lower case, and replacing spaces with underscores
colnames(df) = tolower(str_replace_all(colnames(df), c(' ' = '_')))

# Checking column names.
colnames(df)
```

```
##  [1] "invoice_id"             "branch"
##  [3] "customer_type"          "gender"
##  [5] "product_line"           "unit_price"
##  [7] "quantity"               "tax"
##  [9] "date"                   "time"
## [11] "payment"                "cogs"
```

```
## [13] "gross_margin_percentage" "gross_income"
## [15] "rating"                  "total"
```

- Column names have been changed to lower case and spaces replaced with underscores

Null Values

```
colSums(is.na(df))
```

```
##              invoice_id                  branch           customer_type
##                       0                       0                       0
##                  gender            product_line             unit_price
##                       0                       0                       0
##                quantity                     tax                    date
##                       0                       0                       0
##                    time                 payment                    cogs
##                       0                       0                       0
## gross_margin_percentage            gross_income                  rating
##                       0                       0                       0
##                   total
##                       0
```

- No Null values were found

Duplicates

```
# CHecking for duplicated records
sum(duplicated(df))
```

```
## [1] 0
```

- No duplicates were found

Filter Method

```
# Installing and loading the corrplot package for plotting
#
suppressWarnings(
        suppressMessages(if
                         (!require(corrplot, quietly=TRUE))
                 install.packages("corrplot")))
library(corrplot)

# Getting numeric columns
dfn <- df[,!sapply(df, is.character)]
head(dfn)
```

```
## # A tibble: 6 x 9
##   unit_price quantity   tax time    cogs gross_margin_perce~ gross_income
rating
##        <dbl>    <dbl> <dbl> <time> <dbl>               <dbl>        <dbl>
<dbl>
## 1       74.7        7 26.1  13:08  523.                 4.76        26.1
```

```
9.1
## 2       15.3        5  3.82 10:29   76.4                4.76         3.82
9.6
## 3       46.3        7 16.2  13:23  324.                 4.76        16.2
7.4
## 4       58.2        8 23.3  20:33  466.                 4.76        23.3
8.4
## 5       86.3        7 30.2  10:37  604.                 4.76        30.2
5.3
## 6       85.4        7 29.9  18:30  598.                 4.76        29.9
4.1
## # ... with 1 more variable: total <dbl>

str(dfn)

## tibble [1,000 x 9] (S3: tbl_df/tbl/data.frame)
##  $ unit_price             : num [1:1000] 74.7 15.3 46.3 58.2 86.3 ...
##  $ quantity               : num [1:1000] 7 5 7 8 7 7 6 10 2 3 ...
##  $ tax                    : num [1:1000] 26.14 3.82 16.22 23.29 30.21 ...
##  $ time                   : 'hms' num [1:1000] 13:08:00 10:29:00 13:23:00
20:33:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ cogs                   : num [1:1000] 522.8 76.4 324.3 465.8 604.2 ...
##  $ gross_margin_percentage: num [1:1000] 4.76 4.76 4.76 4.76 4.76 ...
##  $ gross_income           : num [1:1000] 26.14 3.82 16.22 23.29 30.21 ...
##  $ rating                 : num [1:1000] 9.1 9.6 7.4 8.4 5.3 4.1 5.8 8 7.2
5.9 ...
##  $ total                  : num [1:1000] 549 80.2 340.5 489 634.4 ...

# Calculating the correlation matrix
# Start by isolating only relevant columns
dfn <- dfn[,c(1,2,3,5,6,7,8)]
# Correlation matrix
correlationMatrix <- cor(dfn)

## Warning in cor(dfn): the standard deviation is zero

# Find attributes that are highly correlated
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.75)
# Highly correlated attributes
#
highlyCorrelated

## [1] 3 4

names(dfn[,highlyCorrelated])

## [1] "tax"  "cogs"
```

- The columns tax and cogs were found to be highly correlated

```r
# Removing columns with high correlation
dfnn<-dfn[-highlyCorrelated]
head(dfnn)
```

```
## # A tibble: 6 x 5
##   unit_price quantity gross_margin_percentage gross_income rating
##        <dbl>    <dbl>                   <dbl>        <dbl>  <dbl>
## 1       74.7        7                    4.76        26.1     9.1
## 2       15.3        5                    4.76         3.82    9.6
## 3       46.3        7                    4.76        16.2     7.4
## 4       58.2        8                    4.76        23.3     8.4
## 5       86.3        7                    4.76        30.2     5.3
## 6       85.4        7                    4.76        29.9     4.1
```

```r
docs <- dist( as.matrix(dfn), method = "euclidean")
hclust_dist<- as.dist(docs)
hclust_dist[is.na(hclust_dist)]
```

```
## numeric(0)
```

```r
hclust_dist[is.nan(hclust_dist)]
```

```
## numeric(0)
```

```r
sum(is.infinite(hclust_dist)) # THIS SHOULD BE 0
```

```
## [1] 0
```

```r
hclust(hclust_dist, "ward.D2")
```

```
##
## Call:
## hclust(d = hclust_dist, method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : euclidean
## Number of objects: 1000
```

```r
# We can remove the variables with a higher correlation
# and comparing the results graphically as shown below
#
# Removing Redundant Features
#
#Dataset<-docs[-highlyCorrelated]

# Performing our graphical comparison
# ---
#
#par(mfrow = c(1, 2))
#corrplot(correlationMatrix, order = "hclust")
#corrplot(cor(Dataset), order = "hclust")
```

```
#rlang::last_error()
```

Wrapper Method

```
# Importing Libraries
library(clustvarsel)

## Warning: package 'clustvarsel' was built under R version 4.0.5

## Loading required package: mclust

## Warning: package 'mclust' was built under R version 4.0.5

## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.

##
## Attaching package: 'mclust'

## The following object is masked from 'package:purrr':
##
##     map

## Package 'clustvarsel' version 2.3.4

## Type 'citation("clustvarsel")' for citing this R package in publications.

library(mclust)

# Scaling the data
#dfnscaled <- scale(dfn)
# Sequential forward search
#dfnseq = clustvarsel(dfnscaled, G=1:3)
#dfnseq

#rlang::last_error()

#subset_1 = dfn[,dfnseq$Subset]
#mod=Mclust(subset_1,G=1:3)
#summary(mod)
```

Embedded method

```
# Importing libraries
#library(wskm)

# Setting random seed
#set.seed(3)
# Create feature selection model
#library('cluster')
#model <- ewkm(dfn,3,lambda=2, maxiter=1000)
# CLuster plot of principal coomponents
#clusplot(dfn, model$cluster, color=FALSE, Shade=TRUE, label=2, lines=1,
         #main='Cluster of customers')
```