DATA REPROCESSING DIAM

THEDION V. DIAM JR.

2022-12-15

Load Data Sets

The data contains 197 rows and 431 columns with *Failure.binary* binary output.

```
library(readr)
rawd <- read_csv("D:/DIAM/FP-DATA.csv")
```

```
## Rows: 197 Columns: 431
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr   (1): Institution
## dbl (430): Failure.binary, Failure, Entropy_cooc.W.ADC, GLNU_align.H.PET,
Mi...
##
## 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.
```

#=============== Reprocessing the Raw Data ================================#

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ dplyr   1.0.10
## ✓ tibble  3.1.8      ✓ stringr 1.4.1
## ✓ tidyr   1.2.1      ✓ forcats 0.5.2
## ✓ purrr   0.3.5
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(bestNormalize)
```

Check for null and missing values

Using *anyNA()* function, We can determine if any missing values in our data.

```
anyNA(rawd)

## [1] FALSE

#The result shows either *True* or *False*. If True, omit the missing values
using *na.omit()*

#[1] FALSE

#Thus, our data has no missing values.
```

Check for Normality of the Data

We used *Shapiro-Wilk's Test* to check the normality of the data.

```
rd <- rawd%>%select_if(is.numeric)
rd <- rd[,-1]
test <- apply(rd,2,function(x){shapiro.test(x)})
```

To have the list of p-value of all variables, the *unlist()* function is used and convert a list to vector.

```
pvalue_list <- unlist(lapply(test, function(x) x$p.value))

sum(pvalue_list<0.05)  # not normally distributed

## [1] 428

sum(pvalue_list>0.05)  # normally distributed

## [1] 1

test$Entropy_cooc.W.ADC

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.98903, p-value = 0.135

# [1] 428
# [1] 1

#  Thus, we have 428 variables that are not normally distributed and
Entropy_cooc.W.ADC is normally distributed.
```

We use *orderNorm()* function, the *x.t* is the elements of orderNorm() function transformed original data.Using the *Shapiro-Wilk's Test*

```
TRDrawd=rawd[,c(3,5:length(names(rawd)))]
```

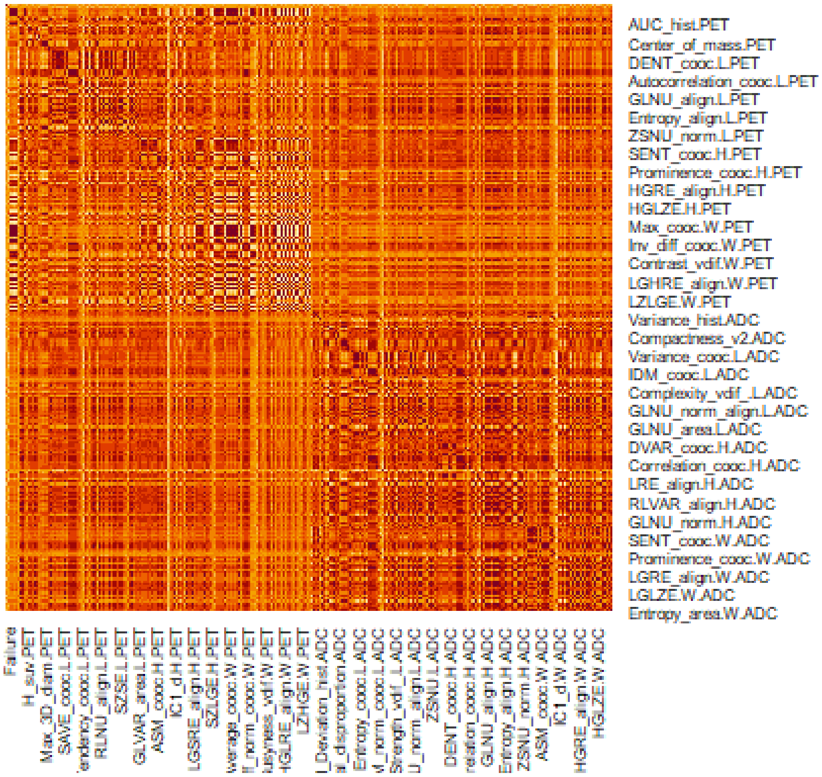
```
TRDrawd=apply(TRDrawd,2,orderNorm)
TRDrawd=lapply(TRDrawd, function(x) x$x.t)
TRDrawd=TRDrawd%>%as.data.frame()
test=apply(TRDrawd,2,shapiro.test)
test=unlist(lapply(test, function(x) x$p.value))
```

#Testing Data

```
sum(test <0.05)  # not normally distributed

## [1] 0

sum(test >0.05)  # normally distributed

## [1] 428

#[1] 0
#[1] 428

# Thus, our data is normally distributed.

rawd[,c(3,5:length(names(rawd)))]=TRDrawd
```

Get the correlation of the whole data expect the categorical variables

```
CorMatrix=cor(rawd[,-c(1,2)])
heatmap(CorMatrix,Rowv=NA,Colv=NA,scale="none",revC = T)
```



#Splitting the Data Split the data into training (80%) and testing (20%).

```
rawd$Institution=as.factor(rawd$Institution)
rawd$Failure.binary=as.factor(rawd$Failure.binary)

splitter <- sample(1:nrow(rawd), round(nrow(rawd) * 0.8))
trainND <- rawd[splitter, ]
testND  <- rawd[-splitter, ]
```

The data frame output of data reprocessing will be converted into to “csv”, which will be used for entire project.

Load new Data

```
Final <- read_csv("D:/DIAM/newdat.csv")

## Rows: 197 Columns: 431
## ── Column specification
────────────────────────────────────────────────────────

## Delimiter: ","
## chr   (1): Institution
## dbl (430): Failure.binary, Failure, Entropy_cooc.W.ADC, GLNU_align.H.PET,
Mi...
##
## 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.

View(Final)
```