```
---
output:
  html_document: default
  pdf_document: default
---
--
title: "Group Project_Data _Processing"
author: "EMMA OO"
date: '2022-06-12'
output: html_document
---

```{r}
library(caret)
library(dplyr)
df <- read.csv("/Users/emmaoo/Desktop/train.csv")
head(df)
```
Drop All Uninformative Columns
```{r}
head(df)
df <- subset(df, select = -
c(Patient.Id,Patient.First.Name,Family.Name,Father.s.name,Institute.Name,
                              Location.of.Institute,Place.of.birth,Test.
1,Test.2,Test.3,Test.5))
head(df,2)
dim(df)
```


Check Unique Values
```{r}
# Check unique values from all columns except Blood.cell.count..mcL. and
White.Blood.cell.count..thousand.per.microliter. since we know there's many
unique values in those columns

#Extract those columns and save under subdf
subdf <- head(select(df, -Blood.cell.count..mcL.,-
White.Blood.cell.count..thousand.per.microliter.,-
White.Blood.cell.count..thousand.per.microliter.))

list_unique <- lapply(subdf, unique)# List with unique values
list_unique
```
Replace "", "_", "Not applicable" values with NA
```{r}
df[df == ""] <- NA
df[df == "-"] <- NA
df[df == "Not applicable"] <- NA
```

Check Missing Values
```{r}
library(naniar)
```
```

```
library(dplyr)
library(caret)
gg_miss_var(df, show_pct = TRUE)
```

Check if missing values were realted to the specific class
```{r}
na_values <- df %>% group_by(Disorder.Subclass) %>% miss_var_summary()
ggplot(na_values, aes(Disorder.Subclass, variable, fill=pct_miss)) +
geom_tile() +theme(axis.text.x = element_text(angle = 90))
```

Data Partitioning

```{r}
set.seed(1)
trainingrows <- createDataPartition(df$Disorder.Subclass, p = 0.80, list =
FALSE)

train <- df[trainingrows,]
test <- df[trainingrows,]

```


Replace Missing Values with Median For Numerical Variables

```{r}
library(dplyr)
library(tidyverse)

Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

#for train data set
train <- train %>% mutate_if(is.numeric, funs(replace(.,is.na(.), median(.,
na.rm = TRUE)))) %>%
  mutate_if(is.character, funs(replace(.,is.na(.), Mode(na.omit(.)))))

# for test data set
test <- test %>% mutate_if(is.numeric, funs(replace(.,is.na(.), median(.,
na.rm = TRUE)))) %>%
  mutate_if(is.character, funs(replace(.,is.na(.), Mode(na.omit(.)))))

sum(is.na(train))
sum(is.na(test))

```

Encoding Yes or No Columns Into Binary Columns
```{r}
library(dplyr)
```

```
train <- train %>%mutate_if(is.character, as.factor)
test <- test %>%mutate_if(is.character,as.factor)

train <- train %>%
  mutate(across(c(Genes.in.mother.s.side,Inherited.from.father,Maternal.gene,Paternal.ge
                  Folic.acid.details..peri.conceptional.,H.O.serious.maternal.illness,
                  H.O.radiation.exposure..x.ray.,H.O.substance.abuse,Assisted.conception
                  History.of.anomalies.in.previous.pregnancies),
~factor(ifelse(.x == "Yes","1","0"))))

test <- test %>%
  mutate(across(c(Genes.in.mother.s.side,Inherited.from.father,Maternal.gene,Paternal.ge
                  Folic.acid.details..peri.conceptional.,H.O.serious.maternal.illness,
                  H.O.radiation.exposure..x.ray.,H.O.substance.abuse,Assisted.conception
                  History.of.anomalies.in.previous.pregnancies),
~factor(ifelse(.x == "Yes","1","0"))))
```

Drop ' and - from the Disorder Subclass for Encoding Purpose
```{r}
#For train data set
train=as.data.frame(lapply(train,gsub,pattern="'",replacement=""))
train=as.data.frame(lapply(train,gsub,pattern="-",replacement=""))
#For test data set
test=as.data.frame(lapply(test,gsub,pattern="'",replacement=""))
test=as.data.frame(lapply(test,gsub,pattern="-",replacement=""))
```

Encoding All Categorical Variables for TRAIN Data set
```{r}

train$Status <- factor(train$Status, levels = c('Alive', 'Deceased'),
labels = c(1,0))
train$Respiratory.Rate..breaths.min. <-
factor(train$Respiratory.Rate..breaths.min., levels = c('Normal (3060)',
'Tachypnea'), labels = c(0,1))
train$Heart.Rate..rates.min <- factor(train$Heart.Rate..rates.min, levels =
c('Normal', 'Tachycardia'), labels = c(0,1))
train$Follow.up <- factor(train$Follow.up, levels = c('High', 'Low'),
labels = c(1,0))
train$Gender <- factor(train$Gender, levels = c('Female',
'Male','Ambiguous'), labels = c(1,2,3))
train$Birth.asphyxia <- factor(train$Birth.asphyxia, levels = c('Yes',
'No','No record','Not available'), labels = c(1,0,2,2))
train$Autopsy.shows.birth.defect..if.applicable. <-
factor(train$Autopsy.shows.birth.defect..if.applicable., levels = c('No',
'None','Yes'), labels = c(0,0,1))

train$Birth.defects <- factor(train$Birth.defects, levels = c('Singular',
'Multiple'), labels = c(1,2))
train$Blood.test.result <- factor(train$Blood.test.result, levels =
c('normal', 'slightly abnormal','abnormal','inconclusive'), labels =
c(0,1,2,3))
```

```
train$Genetic.Disorder <- factor(train$Genetic.Disorder, levels =
c('Mitochondrial genetic inheritance disorders', 'Multifactorial genetic
inheritance disorders','Singlegene inheritance diseases'), labels =
c(1,2,3))
```

Encoding All Categorical Variables for TEST Data set

```{r}
test$Status <- factor(test$Status, levels = c('Alive', 'Deceased'), labels
= c(1,0))
test$Respiratory.Rate..breaths.min. <-
factor(test$Respiratory.Rate..breaths.min., levels = c('Normal (3060)',
'Tachypnea'), labels = c(0,1))
test$Heart.Rate..rates.min <- factor(test$Heart.Rate..rates.min, levels =
c('Normal', 'Tachycardia'), labels = c(0,1))
test$Follow.up <- factor(test$Follow.up, levels = c('High', 'Low'), labels
= c(1,0))
test$Gender <- factor(test$Gender, levels = c('Female',
'Male','Ambiguous'), labels = c(1,2,3))
test$Birth.asphyxia <- factor(test$Birth.asphyxia, levels = c('Yes',
'No','No record','Not available'), labels = c(1,0,2,2))
test$Autopsy.shows.birth.defect..if.applicable. <-
factor(test$Autopsy.shows.birth.defect..if.applicable., levels = c('No',
'None','Yes'), labels = c(0,0,1))

test$Birth.defects <- factor(test$Birth.defects, levels = c('Singular',
'Multiple'), labels = c(1,2))
test$Blood.test.result <- factor(test$Blood.test.result, levels =
c('normal', 'slightly abnormal','abnormal','inconclusive'), labels =
c(0,1,2,3))

test$Genetic.Disorder <- factor(test$Genetic.Disorder, levels =
c('Mitochondrial genetic inheritance disorders', 'Multifactorial genetic
inheritance disorders','Singlegene inheritance diseases'), labels =
c(1,2,3))

head(test)
```

Splitting numerical and categorical predictors for visualization purpose
```{r}
library(dplyr)
num_df <- select_if(train, is.numeric)  # Subset numeric columns with dplyr
cat_df <- select_if(train,is.character)
head(num_df)
head(cat_df)
```

Splitting train data into predictors and outcome
```{r}
trainX <- train[,-34]
```

```
trainy <- train$Disorder.Subclass

testX <- test[,-34]
testy <- test$Disorder.Subclass

```

Change all char and factor into numeric variables
```{r}
trainX <- trainX %>% mutate_if(is.character, as.numeric)
trainX <- trainX %>% mutate_if(is.factor, as.numeric)
```


#Boxplot
```{r}
par(mar=c(10,2,1,1))
boxplot(trainX, las=2, col = c("turquoise","skyblue"))
```

Countplot
```{r}
library(ggplot2)
library(patchwork)
library(cowplot)
Diseases <- trainy
p1 <- ggplot(train, aes(x = Patient.Age, fill = Diseases)) +geom_bar() +
theme_classic()+scale_fill_hue(c=60, l=80)
p2 <- ggplot(train, aes(x = Maternal.gene, fill = Diseases)) +geom_bar()
+theme_classic()+ scale_fill_hue(c=60, l=80)
p3 <- ggplot(train, aes(x = Paternal.gene, fill = Diseases)) +geom_bar()
+theme_classic()+ scale_fill_hue(c=60, l=80)

p1+p2/p3
```

```{r}
library(Hmisc)
hist(trainX)
```

Check highly correlated predictors
```{r}
corr <- cor(trainX)
highcor <- findCorrelation(corr, 0.70)
colnames(train)[highcor]
```
Check Near Zero Variance Predictors and Dropping them
```{r}
trainX <- trainX[,-nearZeroVar(trainX)]
testX <- testX[,-nearZeroVar(testX)]
dim(trainX)
dim(testX)
```

Skewness
```{r}
library(moments)
skewness(trainX)
```