```
""
Datos generales:
Nombre de la entrega: Semana 3 - Actividad 1
Nombre: Rodrigo Rodriguez Rodriguez
Matrícula: A01183284
Materia:Ciencia y analítica de datos (Gpo 10)
Profesor:Jobish Vallikavungal Devassia
Fecha:28/09/2022
"""


"""Parte 1 """
"""
1>. Fundamentos de bases de datos y para ciencia de datos.
Que es una base de datos:
Se cataloga como datos almacenados de forma estructurada y sistemática,
todo esto con el objetivo de poder usados después de forma fácil, las bases
de datos pueden tener cualquier tipo de
información# La ciencia de datos utiliza estas bases de datos para poder
realizar predicción o estudios de forma más
precisa y al conjunto de esta base de datos se les llama almacenes de datos,
2>. Fundamentos de almacenes de datos (Data Warehouse) para ciencia de datos.
Que es un DW son bases de datos que están guardados ya sean viejos o nuevos
con un interés en particular,
estos nos ayudan a poder tomar decisiones y poder con esto tomar decisiones
dependiendo del análisis que nos
arroje al final
Estos datos pueden venir desde otras bases de datos, transacciones o fuentes
siendo inclusive actualizados en tiempo real
Esta información puede modificar un negocio o crear cambios de campaña de
forma casi inmediata con las tecnologías del
hoy en día, de la misma manera tener un almacén de datos puede evolucionar
nuestro negocio y darnos una ventaja
competitiva contra el problema o negocio al que nos enfrentemos
"""


"""Ejercicios"""

import pandas as pd
import numpy as np

"""
X1 Amount of the given credit (NT dollar): it includes both the individual
consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 =
others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).
X6 - X11: History of past payment. We tracked the past monthly payment
records (from April to September, 2005) as follows: X6 = the repayment status
in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 =
the repayment status in April, 2005.
The measurement scale for the repayment status is: -1 = pay duly; 1 = payment
delay for one month; 2 = payment delay for two months; . . .; 8 = payment
delay for eight months; 9 = payment delay for nine months and above.
```

```
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement
in September, 2005; X13 = amount of bill statement in August, 2005; . . .;
X17 = amount of bill statement in April, 2005.
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in
September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid
in April, 2005.
"""

#raw =
r'https://raw.githubusercontent.com/PosgradoMNA/Actividades_Aprendizaje-
/main/default%20of%20credit%20card%20clients.csv'
raw= r'default of credit card clients.csv'
print(raw)
df = pd.read_csv(raw)
# conversion to CSV
# print(df)
df = pd.read_csv(raw, index_col=0)
# print(df)
print(df.isnull().values.any())
# vamos a verificar si los datos no estan
print(df.isnull().any())
# verificamos por linea
# print(df)

# Apliquemos la solucion 1
ds1 = df.copy()
ds1.dropna(inplace=True)
print(ds1.isna().values.any())
print(ds1.isnull().any())  # verificamos por linea
print(ds1)
# nos damos cuenta de que esta solucion no funciona ya que perdemos datos los
cuales son necesarios#
# Verificamos si la tecnica funciona en columnas
Copy = df.copy()
Copy.dropna(axis=1, inplace=True)  # axis 1 is columns / axis 0 is rows.
print(Copy)
# Tampoco funciona ya que perdemos todas las columnas menos una
# utlizamos otra tecnica donde solo dejamos 2 valores de cada nan
treshold = df.copy()
treshold.dropna(how='all', inplace=True)
treshold.dropna(thresh=4, inplace=True)  # en este sistema estamos probando
que pása si pedimos almenos 4 valores
print(treshold)  # esta tecnica nos ayuda ya que perdemos menos elementos sin
embargo aun seguimos perdiendo 1 fila
print(treshold.isna().values.any())
print(treshold.isnull().any())  # verificamos por linea y vemos que aun
perdemos datos
# eliminemos columnas no necesarias
col = df.copy()
col.dropna(thresh=4,   # if there is not 5 nan values, the column will be
eliminated
           axis=1,
           inplace=True)
print(col)
# con esto estams verificando que ninguna columna tiene mas de 5 datos
faltantes  con esto podemos
# proceder a la solucion numero 2
```

```
# Solucion 2 #
ndf=df.copy()
wmx1=ndf.X1.mean()
wmx5=ndf.X5.mean()
print(wmx1,wmx5)
#para fines de esta prueba usaremos los valores de media para los valores x1
, los valores x5
#para los valores x6 en adelante haremos la misma media pero de forma
horizontal entre esas columnas
#esto debido a que nos ayudara a obterner un mejor criterio en las columnas
faltantes
#sustituimos el 1 y el 5 que son en los que aplicaremos la media ya que son
datos que podemos contestar con la media de estos
ndf["X1"].fillna(value = wmx1,
                 inplace = True)#aunque sabems que el X1 esta lleno si la
base de datos llegara a cambiar debemos verificar este punto
ndf["X5"].fillna(value = wmx5,
                 inplace = True)
print(ndf.isna().values.any())
print(ndf.isnull().any())
#con esto podemos ver que ya tenemos la linea X5 con datos representativos

#Como siguiente punto usaremos el median o mediana sin embargo no veo uso
para este data set
#ocupemos el dato de grado educativo
ndf['X3'].fillna(value = ndf.X3.median(),
                 inplace = True)
print(ndf.isna().values.any())
print(ndf.isnull().any())

#para el Siguente Ejercicio usaremos la moda

mm = ndf.X2.mode() # para fines del ejercicio usare el grado de estudios
# esto nos muestra que la mayori a o la moda esta en ir a la universidad
print(mm)
ndf['X2'].fillna(value = mm[0], inplace = True)
print(ndf.isnull().any())

# eliminamos ciertas columnas solo con impute
# ndf.dropna(subset=['X18','X19','X20','X21','X22','X23'], inplace = True)
# print(ndf)
# print(ndf.isnull().any())
# #PARA fines del ejercicio no elimanare filas que esten vacias buscare la
forma de darles un valor por lo cual esto
# se mantendra en coment el ejercicio anterior

# llenemos los valores de una columna especifica con datos

favs = {'X4': ndf.X4.mode()[0], 'X5': ndf['X5'].mean()}
ndf.X4.fillna(ndf.X4.mode()[0], inplace=True)
ndf.X5.fillna(ndf.X5.mean(), inplace=True)
print(ndf)
print(ndf.isnull().any())

####MEJOR USO DE LA  mediana  contra la media###
data = {'Salary':  [28, 30, 30, 35, 37, 40, 400]
}
```

```
adf = pd.DataFrame(data)
print(adf)
desc=adf.describe()
print(desc)

#La mediana se usa con datos que no se ajustan a un distribucuin normal,
debido al lo grande que puede ser su metodo
#La media se usa para conocer parametros exactamente en la mitad de una
muestra para valores que sigan un logica lineal

# como vemos los valores de una colmna en especifico
print(ndf.columns)
print(ndf.columns.sort_values())
print(ndf.loc[2:5 , 'X6':'X12'])
favs = ['X18','X19','X20','X21','X22','X23']

print(ndf.loc[2:5 , favs])
print(ndf.iloc[2:5, [1,2, 3]])
print(ndf.columns)

for i in ndf.columns:
  print(i)

print( ndf.head(4))

print(ndf.X2.unique())
print(df.groupby(['X2', 'X3']).size())# uso de size
print(df[['X2', 'X3']].value_counts())# Uso de counts

ndf2 = ndf.drop(['X18','X19','X20','X21','X22','X23'], axis = 1)#eliminamos
columnas
print(ndf2)

ndf2.rename(columns = {'X1' : 'AMOUNT', 'X2': 'GENDER'}, inplace = True)
print(ndf2)
"""Parte 2"""
print("inicia parte2")
#raw =
r'https://raw.githubusercontent.com/PosgradoMNA/Actividades_Aprendizaje-
/main/default%20of%20credit%20card%20clients.csv'
raw= r'default of credit card clients.csv'
df = pd.read_csv(raw)
df.dropna(how='all', inplace = True)
df.dropna(thresh=11, inplace=True)
df.drop(['Y',],axis=1,inplace=True)
mx=df.X5.mean()
#eliminate 0

df.loc[(df.X2== 0),'X2']= None
df.loc[(df.X3== 0),'X3']= None
df.loc[(df.X4== 0),'X4']= None
df.loc[(df.X5== 0),'X5']= None

mx=df.X5.mean()
df["X5"].fillna(value = mx,inplace = True)
df["X3"].fillna(value = ndf.X3.median(),inplace = True)
df["X2"].fillna(value = df.X2.mode(),inplace = True)
```

```
df["X4"].fillna(value = df.X4.mode(),inplace = True)
df["X6"].fillna(value = 0 ,inplace = True)
df.loc[df.X6 <- 1,'X6']=-1
df.loc[df.X7 <- 1,'X7']=-1
df.loc[df.X8 <- 1,'X8']=-1
df.loc[df.X9 <- 1,'X9']=-1
df.loc[df.X10 <- 1,'X10']=-1
df.loc[df.X11 <- 1,'X11']=-1
#verificamos los valores negativos de los pagos

#verify ceros
df.loc[(df.X12 == 0)&(df.X18==0),'X12']= None
df.loc[(df.X13 == 0)&(df.X19==0),'X13']= None
df.loc[(df.X14 == 0)&(df.X20==0),'X14']= None
df.loc[(df.X15 == 0)&(df.X21==0),'X15']= None
df.loc[(df.X16 == 0)&(df.X22==0),'X16']= None
df.loc[(df.X17 == 0)&(df.X23==0),'X17']= None

df.loc[(df.X12 == 0),'X12']= None
df.loc[df.X13 == 0,'X13']=df.X13-df.X19
df.loc[df.X14 == 0,'X14']=df.X14-df.X20
df.loc[df.X15 == 0,'X15']=df.X15-df.X21
df.loc[df.X16 == 0,'X16']=df.X16-df.X22
df.loc[df.X17 == 0,'X17']=df.X17-df.X23

#verify negatives
df.loc[df.X12 < -1,'X12']= df.X12 *-1
df.loc[df.X13 <- 1,'X13']=df.X13 *-1
df.loc[df.X14<- 1,'X14']=df.X14*-1
df.loc[df.X15<- 1,'X15']=df.X15 *-1
df.loc[df.X16 <- 1,'X16']=df.X16 *-1
df.loc[df.X17 <- 1,'X17']=df.X17 *-1


#verify negatives
df.loc[df.X18 < -1,'X18']= df.X18 *-1
df.loc[df.X19 <- 1,'X19']=df.X19 *-1
df.loc[df.X20<- 1,'X20']=df.X20*-1
df.loc[df.X21<- 1,'X21']=df.X21 *-1
df.loc[df.X22 <- 1,'X22']=df.X22 *-1
df.loc[df.X23 <- 1,'X23']=df.X23 *-1

#Verify values on payments


df.loc[df.X18 ==0 ,'X18']= df.X12-df.X13
df.loc[df.X19 ==0 ,'X19']= df.X13-df.X14
df.loc[df.X20 ==0 ,'X20']= df.X14-df.X15
df.loc[df.X21 ==0 ,'X21']= df.X15-df.X16
df.loc[df.X22 ==0 ,'X22']= df.X16-df.X17
df.loc[df.X23 ==0 ,'X23']= df.X17-df.X18

df.loc[(df.X6 ==0)&(df.X18 > 0) ,'X6']= -1
df.loc[(df.X7 ==0)&(df.X19 > 0) ,'X7']= -1
df.loc[(df.X8 ==0)&(df.X20 > 0) ,'X8']= -1
df.loc[(df.X9 ==0)&(df.X21 > 0) ,'X9']= -1
df.loc[(df.X10 ==0)&(df.X22 > 0) ,'X10']= -1
```

```
df.loc[(df.X11 ==0)&(df.X23 > 0) ,'X11']= -1
df.loc[(df.X6 ==0)&(df.X18 ==0) ,'X6']= df.X6+1
df.loc[(df.X7 ==0)&(df.X19 == 0) ,'X7']= df.X7+1
df.loc[(df.X8 ==0)&(df.X20 ==0) ,'X8']= df.X8+1
df.loc[(df.X9 ==0)&(df.X21 ==0) ,'X9']= df.X9+1
df.loc[(df.X10 ==0)&(df.X22 == 0) ,'X10']= df.X10+1
df.loc[(df.X11 ==0)&(df.X23 ==0) ,'X11']= df.X11+1
#PAGOS ERRONEOS SON DATA ERRONEA
df.loc[(df.X18 ==0),'X18']= None

df.dropna(axis=0,how='any',inplace=True)


print(df.size)
print(df.isnull().any())
print(df)
df.to_csv (r'export_dataframe.csv', index = False, header=True)

"""Parte  3 """


"""
1. ¿Qué datos considero más importantes? ¿Por qué?
Para el estudio de estos datos
La cantidad de dinero prestado
La edad
El grado educativo
Estado marital
El historial de pagos
La razón principal por la cual decidí esto datos es debido a que con estos 4
parámetros podremos saber que cantidad
puede pagar normalmente alguien de cierta edad y grado de estudios y que tan
complicado se puede poner su situación de
 pago como
Como datos secundarios pondríamos cuanto quedaron a deber o en cuanto tiempo
lograron pagar su deuda
2. ¿Se eliminaron o reemplazaron datos nulos? ¿Qué se hizo y por qué?
Si se debe remplazar para que la base de datos sea de utilidad debemos
verificar cada valor y hacer que  la línea final
no tenga líneas con datos vacíos en varias líneas , esto nos ayudara a
conocer los valores reales de deuda, por otro
lado en unos casos tenemos un valor negativo diferente a -1 lo cual significa
que este debe ser cambiado a -1
También verificamos que los valores en la celda de dinero adeudado no sean
menor a 0 ya que eso puede ser un error de
dedo un probelma mas grande para el analis final

3. ¿Es necesario ordenar los datos para el análisis? Sí / No / ¿Por qué?
No debido a que cada caso tiene su propio análisis y no depende de otra fila
4. ¿Existen problemas de formato que deban solucionar antes del proceso de
modelado? Sí / No / Por qué.
No,ya que la base de datos ya estaba normalizada lo cual nos permite hacer
mejoras a la misma
Solo algunos problemas de numeros negativos donde no deben existir

5. ¿Qué ajustes se realizaron en el proceso de limpieza de datos (agregar,
integrar, eliminar, modificar registros
```

```
(filas), cambiar atributos (columnas)?
Se realizaron ajustes diferentes dependiendo el campo para el sexo al no ser
un parámetro importante se tomo la mediana
para rellenar los datos
En el caso del Education se tomo la media
El marital estatus también al ser de importancia media se tomo la media o
promedio
Para la edad de la misma forma se pretende usar la media aritmética
Para las métricas del X6 en adelante se deben usar medidas de programación
más avanzadas ya que dependemos de otras
columnas no solo de una para saber si el valor no esta lo que causa que
usemos fórmulas de comparación don la línea
de al lado
Eliminamos filas que no tenían suficientes datos, o causaban un ruido no
necesario como los pagos erroneos o los pagos fuera de tiempo
este data set queda listo para poder ser usado


"""
```