texto del vínculo

Ciencia de Datos y Almacenamiento de Datos

Actualmente los científicos de datos dedican un 80% del tiempo y datos al preprocesamientos y limpieza de las bases de datos ya que uno de los grandes desafíos de las ciencias de los datos es como tratar esta información con datos faltantes o corrompidos para tener análisis más reales y apegado a los objetivos. Para con ellos transformar la manera en la que las empresas operan de manera más precisa confiable y en tiempo real además de poder crear estrategias predictivas hacia el futuro. Almacenamiento de datos es un repositorio central que se puede utilizar para diferentes análisis y toma de decisiones de grandes montos de datos. La base de datos puede estar compuesta en tres partes el front end la información arrojada, el análisis y procesamiento de estos y el almacenamiento de datos agrupados en fila y columnas.

El análisis de datos tiene múltiples beneficios como: toma decisiones en tendencias o resultados, mejorar la calidad de los datos para una mejor claridad y entendimientos de estos. Existen diferentes tipos de almacenamientos como base de datos, un lago de datos y un almacenamiento de datos dependiendo de el volumen de datos y la calidad de estos.

```
import pandas as pd
import numpy as np

df = pd.read_csv('https://raw.githubusercontent.com/PosgradoMNA/Actividades_Aprendizaje-/main
df
```

ID	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X15	X16	X1
1	20000	2.0	2.0	1.0	24.0	2.0	2.0	-1.0	-1.0	-2.0	...	0.0	0.0	0.

```
#Buscamos si un Valor es faltante
df.isnull().values.any()
```

```
False
```

```
#Buscamos si un dato es faltante
df.isnull().any()
```

```
X1     False
X2     False
X3     False
X4     False
X5     False
X6     False
X7     False
X8     False
X9     False
X10    False
X11    False
X12    False
X13    False
X14    False
X15    False
X16    False
X17    False
X18    False
X19    False
X20    False
X21    False
X22    False
X23    False
Y      False
dtype: bool
```

```
df.isna().values.any()
```

```
True
```

```
df.isna().any()
```

```
X1     False
X2      True
X3      True
X4      True
X5      True
```

```
X6      True
X7      True
X8      True
X9      True
X10     True
X11     True
X12     True
X13     True
X14     True
X15     True
X16     True
X17     True
X18     True
X19     True
X20     True
X21     True
X22     True
X23     True
Y       True
dtype: bool
```

```
df
```

ID	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X15	X16	X1
1	20000	2.0	2.0	1.0	24.0	2.0	2.0	-1.0	-1.0	-2.0	...	0.0	0.0	0.
2	120000	2.0	2.0	2.0	26.0	-1.0	2.0	0.0	0.0	0.0	...	3272.0	3455.0	3261.
3	90000	2.0	2.0	2.0	34.0	0.0	0.0	0.0	0.0	0.0	...	14331.0	14948.0	15549.
4	50000	2.0	2.0	1.0	37.0	0.0	0.0	0.0	0.0	0.0	...	28314.0	28959.0	29547.
5	50000	1.0	2.0	1.0	57.0	-1.0	0.0	-1.0	0.0	0.0	...	20940.0	19146.0	19131.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
29996	220000	1.0	3.0	1.0	39.0	0.0	0.0	0.0	0.0	0.0	...	88004.0	31237.0	15980.
29997	150000	1.0	3.0	2.0	43.0	-1.0	-1.0	-1.0	-1.0	0.0	...	8979.0	5190.0	0.
29998	30000	1.0	2.0	2.0	37.0	4.0	3.0	2.0	-1.0	0.0	...	20878.0	20582.0	19357.
29999	80000	1.0	3.0	1.0	41.0	1.0	-1.0	0.0	0.0	0.0	...	52774.0	11855.0	48944.
30000	50000	1.0	2.0	1.0	46.0	0.0	0.0	0.0	0.0	0.0	...	36535.0	32428.0	15313.

30000 rows × 24 columns

```
# Elimina las filas que tienen datos nulos en este caso pone dato boolean
df.dropna(inplace = True)
df.isna().values.any()
```

```
False
```

```
df
```

ID	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X15	X16	X1
1	20000	2.0	2.0	1.0	24.0	2.0	2.0	-1.0	-1.0	-2.0	...	0.0	0.0	0.
2	120000	2.0	2.0	2.0	26.0	-1.0	2.0	0.0	0.0	0.0	...	3272.0	3455.0	3261.
3	90000	2.0	2.0	2.0	34.0	0.0	0.0	0.0	0.0	0.0	...	14331.0	14948.0	15549.
4	50000	2.0	2.0	1.0	37.0	0.0	0.0	0.0	0.0	0.0	...	28314.0	28959.0	29547.
5	50000	1.0	2.0	1.0	57.0	-1.0	0.0	-1.0	0.0	0.0	...	20940.0	19146.0	19131.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
29996	220000	1.0	3.0	1.0	39.0	0.0	0.0	0.0	0.0	0.0	...	88004.0	31237.0	15980.
29997	150000	1.0	3.0	2.0	43.0	-1.0	-1.0	-1.0	-1.0	0.0	...	8979.0	5190.0	0.
29998	30000	1.0	2.0	2.0	37.0	4.0	3.0	2.0	-1.0	0.0	...	20878.0	20582.0	19357.
29999	80000	1.0	3.0	1.0	41.0	1.0	-1.0	0.0	0.0	0.0	...	52774.0	11855.0	48944.
30000	50000	1.0	2.0	1.0	46.0	0.0	0.0	0.0	0.0	0.0	...	36535.0	32428.0	15313.

29958 rows × 24 columns

Haz doble clic (o ingresa) para editar

Productos pagados de Colab - Cancela los contratos aquí

✓ 0 s se ejecutó 22:57