

## Casos datos nulos

Si los valores nulos de la columna *color* no se cruzan con los nulos de la columna *edad*, se eliminaría el 60 % más el 12%, respectivamente, del total de registro. Para este caso, solamente quedaría para trabajar un 28 % de los datos.

No	color	edad
1	Nulos	
2	Nulos	
3	Nulos	
4	Nulos	
5	Nulos	
6	Nulos	
7		Nulos
8		Nulos
9		
10		

Diagram illustrating the elimination of null values from a dataset. The table shows 10 records. Records 1-6 have null values in the 'color' column (highlighted in yellow). Records 7-8 have null values in the 'edad' column (highlighted in blue). A large blue arrow labeled '60%' points to the 'color' column, indicating the percentage of records to be eliminated due to null values in 'color'. A smaller blue arrow labeled '12%' points to the 'edad' column, indicating the percentage of records to be eliminated due to null values in 'edad'. A final blue arrow labeled '28%' points to the remaining records (9 and 10), indicating the percentage of data that remains after elimination.

Para el caso en que los nulos de la variable *color* coincidan con los de la variable *edad*, como se muestra en la siguiente tabla, se estaría eliminado el 60 %, porcentaje, de igual manera, bastante alto para pensar en eliminación, ya que quedarían solo un 40 % del total de la muestra.

No	color	edad
1	Nulos	
2	Nulos	Nulos
3	Nulos	Nulos
4	Nulos	
5	Nulos	
6	Nulos	
7		
8		
9		
10		

Diagram illustrating the elimination of null values from a dataset where null values in 'color' and 'edad' coincide. The table shows 10 records. Records 1-6 have null values in the 'color' column (highlighted in yellow). Records 2-3 have null values in the 'edad' column (highlighted in blue). A large blue arrow labeled '60%' points to the 'color' column, indicating the percentage of records to be eliminated due to null values in 'color'. A final blue arrow labeled '40%' points to the remaining records (7-10), indicating the percentage of data that remains after elimination.

Al ser muy alta la eliminación, es necesario revisar las demás columnas para determinar si existen valores nulos en menor cantidad. Efectivamente, los porcentajes que se obtuvieron para *marca* y *género* están muy cercanos a 0.

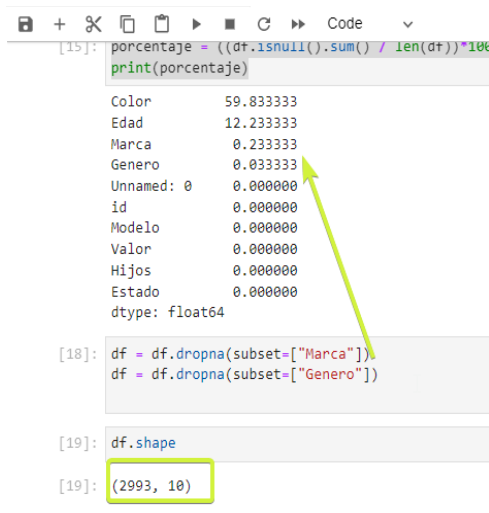
Así que este será el punto de partida para quitar esos pequeños registros, que no son representativos con respecto al total de registros. La estrategia será quitar las filas donde se detecten valores nulos, pero en las columnas indicadas, que para este caso serían *marca* y *género*. Para realizar esa eliminación, se deben usar los siguientes comandos:

```
df = df.dropna(subset=["Marca"])
```

```
df = df.dropna(subset=["Genero"])
```

```
df.shape
```

**Figura 1** Eliminación de nulos por columna



```
[15]: porcentaje = ((df.isnull().sum() / len(df))*100)
      print(porcentaje)

Color      59.833333
Edad       12.233333
Marca       0.233333
Genero      0.033333
Unnamed: 0  0.000000
id          0.000000
Modelo     0.000000
Valor      0.000000
Hijos      0.000000
Estado     0.000000
dtype: float64

[18]: df = df.dropna(subset=["Marca"])
      df = df.dropna(subset=["Genero"])

[19]: df.shape

[19]: (2993, 10)
```

En la Figura 1, el resultado que arroja es la eliminación de solo 7 registros, aunque existía 1 registro adicional para género; esto se debe a que se cruzaron los valores nulos para esas dos columnas, como se explicaba anteriormente con las tablas.

Este primer procedimiento que se realizó sirvió para quitar esos pequeños registros que no alcanzan a superar el 1 % del total de los datos, y que, al quitarlos, no estarían representando ningún riesgo.

Para el caso en que los porcentajes de valores nulos sean pequeños, simplemente se debe ejecutar el comando de eliminación siguiente:

```
df.dropna()
```

El cual eliminará toda fila donde encuentre un valor nulo en cualquiera de las columnas, ya sabemos que para este caso no funcionaría, toda vez que se eliminaría por lo menos un 60 % del total de los registros.

Aún tenemos la columna *color*, con un porcentaje alto de valores nulos que es necesario solucionar. El siguiente comando lo puede usar para optimizar la eliminación de todas las columnas en las cuales el porcentaje de nulos supere el 50 %.

```
porcentaje=50.0
```

```
CantidadMinima=int(((100-porcentaje)/100)*df.shape[0]+1)
```

```
df=df.dropna(axis=1, thresh=CantidadMinima)
```

```
df
```

Donde *porcentaje* es el porcentaje máximo de datos nulos que se permite, y *CantidadMinima* es el cálculo del número mínimo de valores no nulos que debe tener una columna para no eliminarla.

El resultado, como era de esperarse, es la eliminación de la columna *color*, en la que se presentaba un porcentaje por encima del 50% de los valores nulos. (Véase Figura 2)

**Figura 2 Eliminación teniendo en cuenta el porcentaje de nulos**

The screenshot shows a Jupyter Notebook interface. In the code cell, the following code is executed:

```
[24]: porcentaje=50.0
CantidadMinima=int(((100-porcentaje)/100)*df.shape[0]+1)
df=df.dropna(axis=1, thresh=CantidadMinima)
df
```

The output of the code is a preview of the DataFrame. The first few rows are shown, followed by an ellipsis, and then the last few rows. The columns are: Unnamed: 0, id, Modelo, Marca, Valor, Hijos, Genero, Edad, Estado. The 'Edad' column is highlighted in yellow in the original image.

	Unnamed: 0	id	Modelo	Marca	Valor	Hijos	Genero	Edad	Estado
0	0	1	2003	Honda	156298910	2	F	17.0	False
1	1	2	1990	Lotus	128478806	2	F	66.0	True
2	2	3	2003	Mitsubishi	130559812	1	M	31.0	True
3	3	4	1985	Audi	168943347	4	M	10.0	True
4	4	5	1998	GMC	192141187	1	F	67.0	False
...	...	...	...	...	...	...	...	...	...
2995	995	996	2003	Lexus	57235066	1	M	45.0	True
2996	996	997	1992	Dodge	114144343	3	M	38.0	True
2997	997	998	2004	Mazda	122594122	1	M	11.0	True
2998	998	999	1992	Lexus	60274066	2	M	47.0	False
2999	999	1000	1988	Audi	186132184	2	F	NaN	False

At the bottom of the preview, it says "2993 rows x 9 columns".

Hasta este punto, se han utilizado dos estrategias de eliminación; la primera, teniendo en cuenta valores nulos, pero especificando la columna; y la segunda, con la eliminación de columnas que superaran el 50 % de nulos. Dé un vistazo al resultado hasta este momento, para ello, puede utilizar el comando *df.info()*.

**Figura 15 Resultados con la eliminación de nulos**

The screenshot shows a Jupyter Notebook interface. In the code cell, the following code is executed:

```
[25]: df.info()
```

The output of the code is the information of the DataFrame:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2993 entries, 0 to 2999
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Unnamed: 0  2993 non-null  int64
1   id          2993 non-null  int64
2   Modelo     2993 non-null  int64
3   Marca      2993 non-null  object
4   Valor      2993 non-null  int64
5   Hijos      2993 non-null  int64
6   Genero     2993 non-null  object
7   Edad       2626 non-null  float64
8   Estado     2993 non-null  bool
dtypes: bool(1), float64(1), int64(5), object(2)
memory usage: 213.4+ KB
```

The 'Edad' row is highlighted in yellow in the original image.

Como se observa en la Figura anterior, ya casi todas las variables están unificadas en 2993 registros, aunque la edad sigue siendo aún menor, lo que indica que existen valores nulos. La estrategia a seguir para esta columna será la de imputación, que verá a continuación.