Benemérita Universidad Autónoma de Puebla

FCC

Alfredo García

Inteligencia de Negocios

Técnicas de tratamiento de valores nulos

202154423

José Armando Ramos Ramos

Periodo: Primavera 2025

```
#carga desde un archivo .csv sin indice
data = pd.read_csv('listings.csv')
data.head(5)
```

Cargamos el archivo csv sin índice y que solo nos muestre las primeras cinco filas

```
#Corroboramos valores nulos
valores_nulos = data.isnull().sum()
valores_nulos
```

Vemos que variables son las que tienen valores nulos, sin embargo, no nos muestra todos

```
# Identificar y mostrar columnas con valores nulos
valores_nulos = data.isnull().sum()
valores_nulos = valores_nulos[valores_nulos > 0]  # Filtrar solo columnas con nulos

print(valores_nulos)
```

Con ello, podemos ver únicamente las variables que tienen valores nulos, en este caso mayor a cero.

```
#Realizamos una copia del dataframe
data2= data.copy()
```

Hacemos una copia del dataframe para poder corregir los valores nulos

```
data2.info()
```

Vemos la información del dataset, para ver “Dtype” el cual nos dice cual es el tipo de dato y con ello saber que técnica usar.

```
data2["host_response_time"] = data2["host_response_time"].fillna("Estado de manera desconocida")
```

Iniciamos con la variable “host_response_time” el cual es de tipo “object”, por el cual debemos usar un string, para poder llenar los valores nulosm en este caso “Estado de manera desconocida”.

```
#Quinto método de sustitución de valores nulos
#Sustituir valores nulos por valores no nulos hacia adelante "forward fill" ("ffill")
#Filtro por columnas
data2["host_response_rate"] = data2["host_response_rate"].fillna(method="ffill")
```

Utilizamos “ffill”, ya que nos permite ejecutarlo sin límites del tipo de dato, ffill" significa "forward fill". Este método toma el último valor válido (no nulo) anterior a un NaN y lo copia en las posiciones donde hay NaN.

```
#Sexto método de sustitución de valores nulos
#Sustituir valores nulos por valores no nulos hacia atrás backward fill ("bfill")
#Filtro por columnas
data2["host_acceptance_rate"] = data2["host_acceptance_rate"].fillna(method="bfill")
#La variable requirio dos tecnicas, ya que con el bfill, faltaron 8, con un ffill se pudieron quitar los últimos 8 nulos que
quedaban
# data2["host_acceptance_rate"] = data2["host_acceptance_rate"].fillna(method="ffill")
```

Para la variable “host_acceptance_rate” utilizamos el método bfill en pandas (también conocido como "Backward Fill") rellena los valores nulos (NaN) en una columna o DataFrame utilizando el siguiente valor no nulo que se encuentra hacia abajo (es decir, mirando hacia "el futuro"). Sin embargo, nos dejaba con 8 valores que aún seguían nulos, tuve que aplicar un ffill para poder llenar los que faltaban

```
#Cuarto método de sustitución de valores nulos
#Sustituir valores nulos por string en concreto
data2["host_is_superhost"] = data2["host_is_superhost"].fillna("Sin identificar")
```

Si existen valores nulos (NaN) en esa columna, serán reemplazados por el string "Sin identificar". fillna() es una función de pandas que permite reemplazar los valores nulos con un valor específico, en este caso un texto.

```
#Primer método de sustitución de valores nulos
#Sustituir nulos con promedio o media
data2["bathrooms"] = data2["bathrooms"].fillna(round(data["bathrooms"].mean(),1))
```

Para la variable “bathrooms”, fillna() detecta los valores nulos (NaN) en la columna y permite reemplazarlos con un valor específico y redondea el promedio calculado a un decimal.

```
#Quinto método de sustitución de valores nulos "forward ffill" ("ffill")
data2["bathrooms_text"] = data2["bathrooms_text"].fillna(method="ffill")
```

Utilizamos “ffill”, ya que nos permite ejecutarlo sin límites del tipo de dato, ffill" significa "forward fill". Este método toma el último valor válido (no nulo) anterior a un NaN y lo copia en las posiciones donde hay NaN.

```
#Segundo método de sustitución de valores nulos
#Sustituir valores nulos con mediana
data2['bedrooms'] = data2["bedrooms"].fillna(round(data["bedrooms"].median(),1))
```

En este caso median, calcula el promedio (media aritmética) de los valores no nulos en la columna "bedrooms" del DataFrame data.

```
info = data2['review_scores_rating'].dtype
info
```

En este apartado como en data info no me daba el tipo de todos ya que son 50, lo hice de tal manera que poniendo la variable me da el tipo, esto se logra con “.dtype”

```
#Primer método de sustitución de valores nulos
#Sustituir nulos con promedio o media
data2["beds"] = data2["beds"].fillna(round(data["beds"].mean(),1))
```

De igual manera la variable “beds”, fillna() detecta los valores nulos (NaN) en la columna "beds" y permite sustituirlos por el valor proporcionado. Calcula el promedio (media aritmética) de los valores no nulos en la columna "beds".

```
#Quinto método de sustitución de valores nulos
#Sustituir valores nulos por valores no nulos hacia adelante "forward fill" ("ffill")
#Filtro por columnas
data2["price"] = data2["price"].fillna(method="ffill")
```

Utilizamos “ffill”, ya que nos permite ejecutarlo sin límites del tipo de dato, ffill" significa "forward fill". Este método toma el último valor válido (no nulo) anterior a un NaN y lo copia en las posiciones donde hay NaN en la variable “price”.

```
#Sexto método de sustitución de valores nulos
#Sustituir valores nulos por valores no nulos hacia atrás backward fill ("bfill")
#Filtro por columnas
data2["has_availability"] = data2["has_availability"].fillna(method="bfill")
```

Para la variable “has_availability” utilizamos el método bfill en pandas (también conocido como "Backward Fill") rellena los valores nulos (NaN) en una columna o DataFrame utilizando el siguiente valor no nulo que se encuentra hacia abajo (es decir, mirando hacia "el futuro").

```
#Cuarto método de sustitución de valores nulos
#Sustituir valores nulos por string en concreto
data2["first_review"] = data2["first_review"].fillna("2011-06-19")
```

Para este caso, en el fillna le puse un string, que es una fecha, los valores que sean nulos serán cambiados por le fecha que establece.

```
#Cuarto método de sustitución de valores nulos
#Sustituir valores nulos por string en concreto
data2["last_review"] = data2["last_review"].fillna("2024-12-05")
```

Para este caso, en el fillna le puse un string, que es una fecha, los valores que sean nulos serán cambiados por le fecha que establece.

```
#Segundo método de sustitución de valores nulos
#Sustituir valores nulos con mediana
data2['review_scores_rating'] = data2["review_scores_rating"].fillna(round(data["review_scores_rating"].median(),1))
```

En este caso median, calcula el promedio (media aritmética) de los valores no nulos en la columna "bedrooms" del DataFrame data.

```
#Primer método de sustitución de valores nulos
#Sustituir nulos con promedio o media
data2["review_scores_accuracy"] = data2["review_scores_accuracy"].fillna(round(data["review_scores_accuracy"].mean(),1))
```

De igual manera fillna() detecta los valores nulos (NaN) en la variable "review_scores_accuracy" y permite sustituirlos por el valor proporcionado. Calcula el promedio (media aritmética) de los valores no nulos en la columna dada.

```
#Quinto método de sustitución de valores nulos
data2["review_scores_cleanliness"] = data2["review_scores_cleanliness"].fillna(method="ffill")
```

Utilizamos “ffill”, ya que nos permite ejecutarlo sin límites del tipo de dato, ffill" significa "forward fill". Este método toma el último valor válido (no nulo) anterior a un NaN y lo copia en las posiciones donde hay NaN en la variable “price”.

```
#Sexto método de sustitución de valores nulos
data2["review_scores_checkin"] = data2["review_scores_checkin"].fillna(method="bfill")
```

Para la variable “review_scores_checkin” utilizamos el método bfill en pandas (también conocido como "Backward Fill") rellena los valores nulos (NaN) en una columna o DataFrame utilizando el siguiente valor no nulo que se encuentra hacia abajo (es decir, mirando hacia "el futuro").

```
data2["review_scores_checkin"] = data2["review_scores_checkin"].fillna(round(data["review_scores_checkin"].mean(),1))
```

Como en los demás casos ocupamos “. mean” que calcula el promedio (media aritmética) de los valores no nulos en la columna dada.

```
# Identificar y mostrar columnas con valores nulos
valores_nulos = data2.isnull().sum()
valores_nulos = valores_nulos[valores_nulos > 0]  # Filtrar solo columnas con nulos

print(valores_nulos)
```

```
✓ 0.0s
eries([], dtype: int64)
```

Una vez terminado nos aparece de la siguiente manera, lo cual indica que ya no tenemos valores nulos en el dataset

```
#Identificar valores nulos por dataframe
valores_nulos = data2.isnull().sum().sum()
valores_nulos
```

```
✓ 0.0s
p.int64(0)
```

Igual comprobado del dataframe.

```
#Identificar valores nulos por dataframe
valores_nulos = data2.isnull().sum()
valores_nulos
```

Corroboramos en los demás que igual no haya nulos.

```
beds                                            0
amenities                                       0
price                                           0
minimum_nights                                  0
maximum_nights                                  0
...
calculated_host_listings_count_entire_homes     0
calculated_host_listings_count_private_rooms    0
calculated_host_listings_count_shared_rooms     0
reviews_per_month                               0
dtype: int64
```

```
#Convertir DataFrame a CSV
data2.to_csv("listing_pais_sin_nulos.csv")
```

Por último, lo convertimos a csv.