Tecnológico de Monterrey

Analítica de datos y herramientas de inteligencia artificial I

Grupo 101

Selección y Limpieza de Datos

Diego Antonio Oropeza Linarte | A01733018

Evidencia

14 de Septiembre del 2025

Introducción:

Durante las pasadas 5 semanas en la materia, tuvimos la tarea de elegir una ciudad y extraer un documento cvs el cual provee información sobre el tipo de propiedades que se ofrecen en la ciudad, así como información general respecto a las ubicaciones en las que se encuentran y datos sobre el host. Elegí la ciudad de Milán, Italia ya que es una ciudad en la cual tuve la oportunidad de vivir por unos meses por lo que considero que esto sería una ventaja al poder tener la oportunidad de analizar la información y poder sacar conclusiones de esta.

1.Eliminación de columnas y variables

Respecto a la etapa de la selección y limpieza de datos fue la primera ya que el objetivo de esta es elegir las columnas que serán necesarias para el estudio de datos que terminaremos haciendo, el documento de Airbnb cuenta con 79 columnas diferentes con información cuantitativa y cualitativa. En mi caso, elegí de ambas ya que considero que es muy importante elegir principalmente las que puedan describir las características con las que cuenta el lugar e información sobre el host. A final se descartaron las más irrelevantes para tener un total de 50 columnas para evaluar.

```
#Filas combinadas y Todas las columnas
df1= df.iloc[ :,list(range(2, 8)) + list(range(9, 19)) + list(range(21, 24)) + [26] + [28] + list(range(30, 34))
+ [35] + [38] + list(range(40, 43)) + list(range(50, 57)) + list(range(61, 66)) + list(range(67, 73)) + [74] + [78] ]
df1.info()

print(df1)
```

Después de tener las variables definidas de acuerdo con su propósito, con el siguiente código, se hizo un proceso de indexación el cual tiene la finalidad de acceder a elementos individuales que se encuentran en una estructura de datos, en este caso el cvs.

2. Restablecer los índices

En este caso no se consideró restablecer los índices por unos más específicos, en lo personal considero que todas las variables seleccionadas de Airbnb son lo suficientemente especificas como para hacer un análisis claro y concreto, a pesar de que algunas se parecen mucho, tienen un identificador único lo cual ayuda de igual manera a poder compararlas entre sí.

3. Seleccionar o filtrar los registros

El propósito que le estoy dando a este trabajo es seleccionar y filtrar las características que para mi seria obligatorio que un host tuviera para que me hospedara ahí, tomando en cuenta principalmente el costo, la disponibilidad, las cosas con las que cuenta el lugar y el servicio del host.

Ya teniendo las 50 variables seleccionadas, hay muchas únicas como el ID o la descripción del lugar que no es tan importante como otras, un ejemplo de estas puede ser que tenga de una a dos camas, que el precio sea accesible o que el host conteste rápido a las solicitudes.

Al imprimir el csv como DataFrame, se muestra una columna llamada unamed la cual elimine con el siguiente código.

```
df1 = df.drop(columns=["Unnamed: 0"])
df.head(2)
```

Para poder filtrar los valores de la columna price tuvimos que convertir los valores a float y quitar las comas (,) el signo de USD ($) para poder proseguir con el filtrado de las columnas.

```
#Eliminar un signo de una columna
df1['price']= df1['price'].str.replace('$','')
df1['price']= df1['price'].str.replace(',','')
 0.0s


#Conversion de  tipo de datode columna de tipo Object a Float
df1['price']= df1['price'].astype(float)
```

Ya teniendo completamente limpio el DataFrame con las 50 columnas elegidas anteriormente, filtre las variables con los siguientes condicionales:

```
#1. Los host que se hayan registrado antes del 31 de dic del 2022.
#Filtro por objeto
filtro=df1[df1["host_since"] < "2022-12-31"]

#Los registros de los host que responden “a más tardar en 1 día”
#Filtro por objeto
filtro=df1[(df1["host_response_time"] == "within a day") | (df1["host_response_time"] == "within an hour") | (df1["host_response_time"] == "within a few hours")]

#Que sea casa o apartamento completo o un cuarto privado.
#Filtro o
filtro=df1[(df1["room_type"] == "Entire home/apt") | (df1["room_type"] == "Private room")]

#Que tengan maximo 2 camas.
#Filtro por comparacion
filtro=df1[df1["beds"] <=2]

#Que el host este catalogado como superhost.
#Filtro por comparacion
filtro=df1[df1["host_is_superhost"] == "t"]

#Que tenga 1 baño.
#Filtro por comparacion
filtro=df1[df1["bathrooms"] == 1]

#Que el costo sea de $300 a $1000
#Filtro Y
filtro=df1[(df1["price"] >=300) & (df1["price"] <=1000)]

filtro.head(2)
```

- Los hosts que se hayan registrado antes del 31 de dic del 2022.
- Los registros de los hosts que responden “a más tardar en 1 día”
- Que sea casa o apartamento completo o un cuarto privado.
- Que tengan maximo 2 camas.
- Que el host este catalogado como superhost.
- Que tenga 1 baño.
- Que el costo sea de $300 a $1000

Esto conforme a las características que me interesan principalmente, hablando sobre los métodos de filtrado, los principales fueron el filtro and, or y el de objeto, dada la interpretación de la información y las necesidades que tenemos en esta etapa del proyecto.

El ultimo paso de este es convertir los cambios en un archivo csv llamado “Filtrado.csv”.

```
#Convertir archivo filtrado a CSV
filtro.to_csv("Filtrado.csv")
```
✓ 0.1s

4. Remplazo y eliminación de valores nulos

Ya teniendo los filtros, convertimos el csv con las nuevas actualizaciones en un data frame y observamos cuantos datos nulos existen con el siguiente código.

```
#Identificar valores nulos por columna
valores_nulos=data1.isnull().sum()
pd.set_option("display.max_rows", None)       # Todas las filas
valores_nulos
```

Al ver que tenemos varios valores nulos en la mayoría de las columnas que se evaluaran, necesitamos agregar valores los cuales sean capaces de disminuir las posibilidades de conclusiones erróneas, la lógica que use en este proceso es poner lo mínimo posible que puede tener el lugar ofrecido en Airbnb, un ejemplo de esto puede ser que el lugar debe tener mínimo un baño disponible en este, en caso de que falt información del host o la descripción, en algunos casos se pondrá que no hay información, a pesar de que esto no especifica nada, terminara remplazando a nuestros valores nulos en las bases de datos.

Los valores fueron remplazados de la siguiente manera:

```
#description
data1["description"] =data1["description"].fillna("No description available")

#Neighborhood overview
data1["neighborhood_overview"] =data1["neighborhood_overview"].fillna("No description available")

#Host Location
data1["host_location"] =data1["host_location"].fillna("Milan, Italy")

#host about
data1["host_about"] =data1["host_about"].fillna("No description available")

#response time
data1["host_response_time"] =data1["host_response_time"].fillna("N/A")

#response rate
data1["host_response_rate"] =data1["host_response_rate"].fillna("N/A")

#acceptance rate
data1["host_acceptance_rate"] =data1["host_acceptance_rate"].fillna("N/A")

#super host
data1["host_is_superhost"] =data1["host_is_superhost"].fillna("f")

#host neighbourhood
data1["host_neighbourhood"] =data1["host_neighbourhood"].fillna("Not defined")

#availability
data1["has_availability"] =data1["has_availability"].fillna("f")

#first review
data1["first_review"] =data1["first_review"].fillna("No date")

#last review
data1["last_review"] =data1["last_review"].fillna("No date")

#license
data1["license"] =data1["license"].fillna("Not included")
```

✓ 0.0s

```
#beds
data1["beds"]= data1["beds"].fillna(1)
data1.head(5)

#review score
data1["review_scores_rating"]= data1["review_scores_rating"].fillna(0)
data1.head(5)


#review score cleanliness
data1["review_scores_cleanliness"]= data1["review_scores_cleanliness"].fillna(0)
data1.head(5)

#review score check-inn
data1["review_scores_checkin"]= data1["review_scores_checkin"].fillna(0)
data1.head(5)

#review score communication
data1["review_scores_communication"]= data1["review_scores_communication"].fillna(0)
data1.head(5)

#review score location
data1["review_scores_location"]= data1["review_scores_location"].fillna(0)
data1.head(5)

#review score location
data1["review_scores_value"]= data1["review_scores_value"].fillna(0)
data1.head(5)

#review scores per month
data1["reviews_per_month"]= data1["reviews_per_month"].fillna(0)
data1.head(5)
```

✓ 0.0s

Dada la naturaleza de las columnas y de la información, para no afectar los datos decidí usar la sustitución de valores nulos por un string o valor numérico en concreto dependiendo del tipo de información que se tenía en cada columna.

Al final teniendo 0 en todas las columnas, este lo convertí en cvs a través de Python.

```
#Convertir archivo filtrado a CSV
data1.to_csv("Valores_Nulos.csv")
```
✓ 0.0s

6. Valores Atípicos

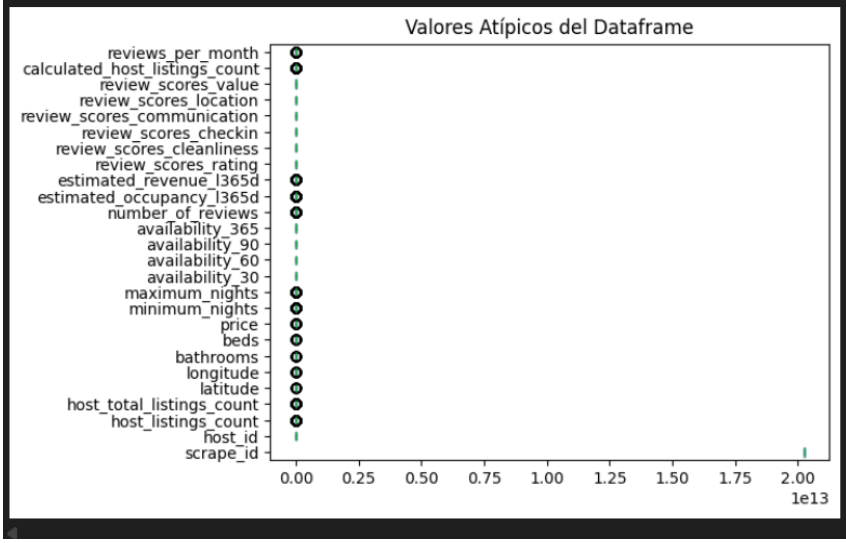
Después de haber eliminado los valores nulos de nuestra base y seguir todos los pasos anteriores, el siguiente paso es identificar los valores atípicos.

Al haber importado las librerías junto con la base y asegurarnos que no hubiera ningún valor nulo, lo siguiente es determinar que valores serán los cuantitativos y cualitativos de acuerdo con lo que nos aparece en nuestra base , con eso podemos proseguir a crear nuestra grafica de caja de acuerdo con el DataFrame de Milán, quedando de la siguiente manera:

```
cuantitativas= data1.select_dtypes(include=["number"])
cualitativas = data1.select_dtypes(include=["object", "category"])
```
✓ 0.0s

```
#Realizamos diagrama de caja o bigote de cada columna del dataframe
fig = plt.figure(figsize =(15, 8))
cuantitativas.plot(kind='box', vert=False)
plt.title("Valores Atípicos del Dataframe")
plt.show() #dibujamos el diagrama
```
✓ 1.4s

```
<Figure size 1500x800 with 0 Axes>
```



Al ver que todo este en orden, el siguiente paso es realizar la desviación estándar. La desviación estándar en Python tiene las funciones de que el número que visualiza es el límite superior permitido para esa columna, queSi un valor en esa columna es mayor que ese límite, se consideraría un valor atípico (outlier) y que probablemente también tienes otra serie con los límites inferiores permitidos (y ahí se ven valores negativos).

Al final esta queda de la siguiente manera:

```
#Método aplicando desviación estandar. Encuentro los valores extremos
y=cuantitativas
Limite_Superior= y.mean() + 3*y.std()
Limite_Inferior= y.mean() - 3*y.std()
print("Limite superior permitido", Limite_Superior)
print("Limite inferior permitido", Limite_Inferior)
```

✓ 0.0s

```
Limite superior permitido scrape_id                      2.025062e+13
host_id                            9.122088e+08
host_listings_count                2.407979e+02
host_total_listings_count          4.544187e+02
latitude                           4.552089e+01
longitude                          9.260303e+00
bathrooms                          4.176673e+00
beds                               8.222687e+00
price                              1.145816e+03
minimum_nights                     6.873836e+01
maximum_nights                     1.614702e+03
availability_30                    4.786453e+01
availability_60                    9.559530e+01
availability_90                    1.413827e+02
availability_365                   5.335426e+02
number_of_reviews                  2.098432e+02
estimated_occupancy_l365d          2.778975e+02
estimated_revenue_l365d            1.474651e+05
review_scores_rating               9.794806e+00
review_scores_cleanliness          9.850629e+00
review_scores_checkin              9.892765e+00
review_scores_communication        9.930960e+00
review_scores_location             9.891748e+00
review_scores_value                9.596564e+00
calculated_host_listings_count     2.056223e+02
...
review_scores_value               -2.744425e+00
calculated_host_listings_count    -1.572613e+02
reviews_per_month                 -3.522514e+00
dtype: float64
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Obtenemos los outliers de nuestra base con la siguiente formula. Es muy importante poder tomar estos en cuenta en nuestro proceso ya que estos son números los cuales se alejan mucho del conjunto de datos por lo que estos pueden afectar el análisis de cualquier base, es de suma importancia eliminarlos o tratarlos para una mejor resolución en los resultados.

```
#Obtenemos datos y los outliers se convierten en nulos en el DataFrame
data3= cuantitativas[(y<=Limite_Superior)&(y>=Limite_Inferior)]
data3.head(1)
```

✓ 0.0s

Posteriormente nos damos cuenta que con este proceso se generan varios valores nulos, los cuales se tiene que limpiar para poder concluir con el proceso, en este caso los estamos anulando con los valores del promedio en donde cualquiera que no tenga un valor, ahora será sustituido por el promedio antes mencionado.

```
#Reemplazamos valores atípicos (nulos) del dataframe con "mean"
#Realizamos una copia del dataframe
data_clean=data3.copy()
data_clean=data_clean.fillna(round(data3.mean(),1))
data_clean.head(1)
```
✓ 0.0s

Otra forma de realizar este proceso es a través del método de cuartiles este siendo de los más comunes para tratar los outliers, en donde se puede realizar el análisis, tomando en cuenta un rango normal basada en la dispersión de cada columna.

Conclusión:

Tomando en cuenta todas las etapas que hemos aplicado a la base de datos de Airbnb durante estas 5 semanas considero que es necesario saber el propósito y bajo que circunstancias se debe de usar cada uno de estos métodos, durante clase, tuvimos la oportunidad de ver múltiples códigos en cada una de nuestras enseñanzas, creo que el hoy pude aplicar de forma efectiva cada una de estas herramientas para poder tratar la información de una mejor manera en donde estuve consiente que en caso de no hacerlo, esto podría cambiar el rumbo de las conclusiones o respuestas que busco responder en base a los datos que se me presentan.

Puedo decir que durante este proyecto pude usar las herramientas conforme a mi conveniencia en donde con ayuda de la creatividad y la intriga pude experimentar y ver como los datos y los resultados cambiaban en base a las decisiones que se toman en el camino, es muy importante que cualquier persona que busque analizar de forma efectiva y responder preguntas respecto a cualquier tema, sea consciente que para esto se necesita tener un entendimiento previo de que es lo que se busca y que es lo que te pueden ofrecer las diferentes herramientas que se nos presentan hoy en día.