CORPORACION UNIVERSITARIA AUTÒNOMA DEL CAUCA

PREPROCESAMIENTO

ENTREGABLE Nº3 TALLER

CRISTIAN CAMILO MARTINEZ CORDOBA

ELECTIVA II

INGENIERÌA ELECTRONICA

MARZO 2025

TALLER

Seguir los siguientes pasos, con el propósito de realizar el pre procesamiento a un dataset de su preferencia.

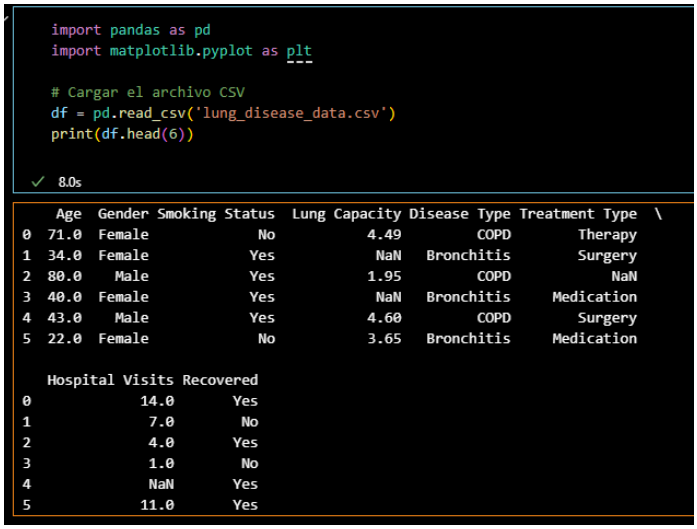
- Descargar dataset en fuentes como Datos Abiertos Colombia, Find Open Datasets and Machine Learning Projects | Kaggle, o la fuente que usted considere adecuada, debe referenciar en el documento el origen
- Identificar problemas en los datos mencionados anteriormente.
- Aplicar limpieza con Pandas.
- Guardar y visualizar el dataset limpio.

Entregables: Documento con portada, la solución a cada inciso con su respectiva captura de pantalla, las conclusiones y la bibliografía necesaria para el desarrollo en formato .PDF. Enlace al repositorio público con los archivos en formato. ipynb y csv generados.

DESARROLLO

- El dataset utilizado en el desarrollo de este taller tiene como nombre: “Conjunto de datos sobre enfermedades pulmonares” extraido de la fuente Kaggle. Este conjunto de datos reúne información detallada sobre pacientes con diversas enfermedades pulmonares, proporcionando una visión integral de su perfil clínico. Incluye datos demográficos como edad y género, permitiendo analizar su distribución en distintos grupos. También registra el estado de tabaquismo de los pacientes, un factor clave en enfermedades pulmonares. Además, mide la capacidad pulmonar para evaluar la gravedad de la afección y especifica el tipo de enfermedad diagnosticada, como EPOC o bronquitis. El conjunto de datos también detalla los tratamientos recibidos, que pueden incluir terapia, medicación o cirugía, así como la frecuencia de visitas al hospital para su manejo. Finalmente, se incluye el estado de recuperación de los pacientes tras el tratamiento, proporcionando información sobre la efectividad de las intervenciones médicas.

- Este código carga y visualiza los primeros registros de un conjunto de datos sobre enfermedades pulmonares. Primero, importa las bibliotecas pandas y matplotlib.pyplot para el manejo de datos y visualización. Luego, utiliza pd.read_csv() para leer el archivo y almacenarlo en un DataFrame llamado df. Finalmente, con df.head(6), muestra las primeras seis filas del dataset, permitiendo una vista rápida de su estructura y contenido.



```
import pandas as pd
import matplotlib.pyplot as plt

# Cargar el archivo CSV
df = pd.read_csv('lung_disease_data.csv')
print(df.head(6))
```

✓ 8.0s

	Age	Gender	Smoking Status	Lung Capacity	Disease Type	Treatment Type	Hospital Visits	Recovered
0	71.0	Female	No	4.49	COPD	Therapy	14.0	Yes
1	34.0	Female	Yes	NaN	Bronchitis	Surgery	7.0	No
2	80.0	Male	Yes	1.95	COPD	NaN	4.0	Yes
3	40.0	Female	Yes	NaN	Bronchitis	Medication	1.0	No
4	43.0	Male	Yes	4.60	COPD	Surgery	NaN	Yes
5	22.0	Female	No	3.65	Bronchitis	Medication	11.0	Yes

Posterior a esto con el siguiente fragmento de código se pueden contar cuantos valores nulos hay en cada columna, obteniendo así un valor de 300 por columna, como se observa a continuación.

```
valores_nulos = df.isnull().sum()

print(valores_nulos)
```

✓ 0.0s Abrir "valores_nulos" en Data Wrangler

```
Age                300
Gender             300
Smoking Status     300
Lung Capacity      300
Disease Type       300
Treatment Type     300
Hospital Visits    300
Recovered          300
dtype: int64
```

De la misma manera, este código elimina las filas duplicadas del DataFrame df. La función drop_duplicates() crea una nueva versión del DataFrame sin registros repetidos y la almacena en df_SinduplicadosCsv. De esta manera, se garantiza que cada fila sea única dentro del conjunto de datos. Cabe resaltar que no se encontraron datos duplicados ya que la cantidad de filas de df_SinduplicadosCsv fue la misma del df inicial.

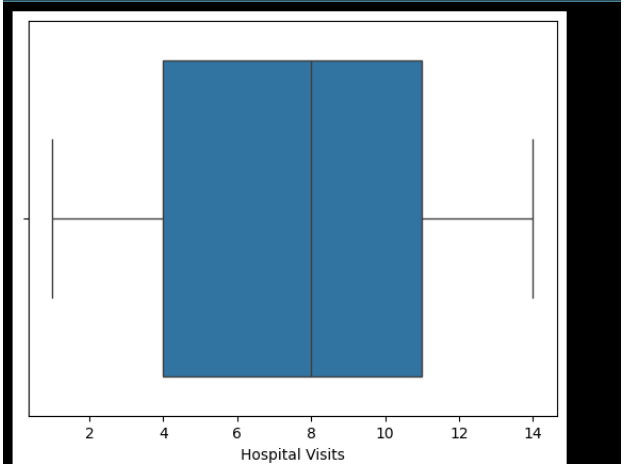
```
df_SinduplicadosCsv = df.drop_duplicates()
```

✓ 0.0s

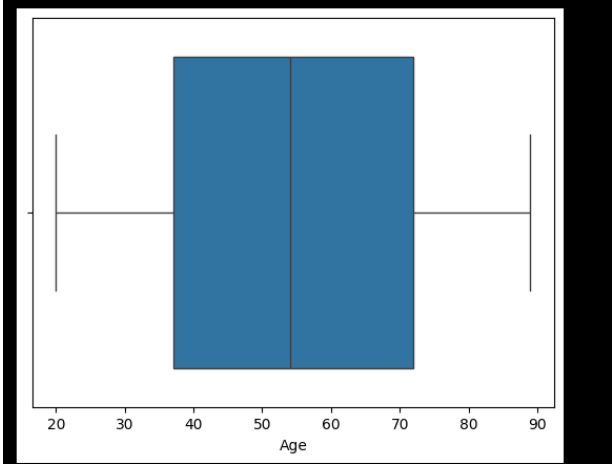
Para la siguiente búsqueda de valores atípicos se decidió tener en cuenta las columnas “Hospital Visits” y “Age”, esto fue posible ya que el código utiliza seaborn y matplotlib.pyplot para visualizar la distribución del número de visitas al hospital en el conjunto de datos. Primero, importa las bibliotecas necesarias. Luego, sns.boxplot(x=df["Hospital Visits"]) genera un diagrama de caja (boxplot) que muestra la distribución, valores atípicos y rango de las visitas hospitalarias registradas en la columna "Hospital Visits". Finalmente, plt.show() muestra el gráfico. Lo mismo con la columna “Age”.

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=df["Hospital Visits"])
plt.show()
```

✓ 0.1s



```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=df["Age"])
plt.show()
```
✓ 0.0s



Como se pudo observar en los 2 diagramas anteriores, para las columnas mencionadas no se obtuvieron valores atípicos o fuera de lo común todos los datos estuvieron apuntando siempre a valores cercanos entre sí. De acuerdo a esto, en el siguiente código se decide aplicar así no sea necesario una corrección para valores atípicos en la variable "Age" como se observa a continuación.

```
dfEdades = df[df['Age'] < 100]
```
✓ 0.0s

El siguiente código convierte a minúsculas. Primero, define la lista columnas_a_minusculas, que contiene los nombres de las columnas a modificar. Luego, usa df[columnas_a_minusculas].apply(lambda x: x.str.lower().str.strip()), donde str.lower() convierte los valores a minúsculas y str.strip() elimina espacios extra. Esto ayuda a mantener la consistencia en los datos y evitar problemas al analizarlos.

```
columnas_a_minusculas = ["Gender", "Smoking Status", "Disease Type", "Treatment Type", "Recovered"]

df[columnas_a_minusculas] = df[columnas_a_minusculas].apply(lambda x: x.str.lower().str.strip())
```

Obteniendo como resultado que se eliminen las mayúsculas en todas las columnas donde había texto.

	# Age	Gender	Smoking Status	# Lung Capacity	Disease Type
	Ausente: 0 (0%) Distinto: 70 (2%) Min 20.0 Max 89.0	Ausente: 0 (0%) Distinto: female male 50% 50%	Ausente: 0 (0%) Distinto: 2 (<1%) yes no 51% 49%	Ausente: 0 (0%) Distinto: 500 (15%) Min 1.0 Max 6.0	Ausente: Distinto: copd bronchitis asthma Otros
0	71.0	female	no	4.49	copd
5	22.0	female	no	3.65	bronchitis
6	41.0	male	yes	2.98	asthma
8	21.0	female	no	4.05	lung cancer
9	49.0	male	yes	5.7	pneumonia
10	57.0	female	yes	4.36	bronchitis
11	21.0	male	no	1.37	pneumonia
12	83.0	male	no	3.3	pneumonia

De la misma manera, para eliminar los valores Nulos para obtener un .csv limpio se debe ejecutar el siguiente fragmento de código.

```
df = df.dropna()
```

Obteniendo como resultado un gran cambio ya que en el dataframe inicial se tenía una cantidad de filas de 5200 y una vez aplicada esta línea de código se pasaron a obtener 3236.

Como paso final se realiza la exportación del archivo una vez se haya hecho cada arreglo que a lo largo de este documento se evidencio, con el siguiente fragmento de código.

```
df.to_csv("datos_limpios.csv", index=False)
```

- **CONCLUSIONES**

Se puede concluir que los procesos como la eliminación de duplicados y la estandarización de texto aseguran que la información sea más confiable y precisa, evitando errores en el análisis.

Visualizar datos mediante gráficos como el boxplot permite identificar patrones, valores atípicos y distribuciones clave, ayudando a comprender mejor la información disponible.

Contar con datos limpios y bien estructurados facilita su manipulación y análisis, reduciendo errores y mejorando la eficiencia en la extracción de información valiosa.

- **REFERENCIAS**

https://www.kaggle.com/datasets/samikshadalvi/lungs-diseases-dataset/data