

OBESITY

PEDRO HERRERA DIAZ

INTROCCION A CIENCIA DE DATOS

JAIME ALEJANDRO ROMERO SIERRA

28 DE NOVIEMBRE DE 2025

INTRODUCCION

La obesidad es un problema importante porque afecta tanto la salud física como la mental. Aumenta el riesgo de enfermedades como diabetes, hipertensión y problemas del corazón. También puede causar problemas emocionales, como baja autoestima y discriminación. Además, genera costos altos en tratamientos y atención médica. Estudiarla ayuda a entender sus diferentes causas —como hábitos, genética y ambiente— y a crear mejores programas de prevención y educación. Abordar este problema puede mejorar la calidad de vida de las personas y proteger la salud de las futuras generaciones.

- Nombre del archivo: ObesityDataSet_raw_and_data_synthtic.csv
- Fuente: El dataset proviene originalmente del Repositorio de Machine Learning de la UCI (University of California, Irvine), comúnmente utilizado para problemas de clasificación y estimación de niveles de obesidad. En el código, los datos se cargan desde un repositorio remoto en GitHub.
- Contexto: Los datos fueron generados a partir de una encuesta realizada en países de México, Perú y Colombia, recopilando información sobre hábitos alimenticios y condición física.

Volumen de Datos (Cantidad)

- Registros (Filas): El dataset cuenta con 2111 registros (individuos encuestados).
- Variables (Columnas): Contiene 17 atributos o características por registro.

Principales Características El dataset combina datos numéricos y categóricos. A continuación se desglosan las variables contenidas:

- Variable Objetivo (Target):
 - NOBES: Nivel de obesidad del individuo. Se clasifica en 7 categorías que van desde "Insufficient_Weight" (Bajo peso), "Normal_Weight" (Peso normal), hasta diferentes niveles de Sobrepeso y "Obesity_Type_III" (Obesidad Tipo 3).
- Datos Demográficos y Antropométricos:
 - Gender: Género (Femenino/Masculino).
 - Age: Edad.
 - Height: Altura (en metros).
 - Weight: Peso (en kilogramos).
 - family_history_with_overweight: Historial familiar de sobrepeso (sí/no).
- Hábitos Alimenticios:

- FAVC: Consumo frecuente de alimentos altos en calorías.
- FCVC: Frecuencia de consumo de vegetales.
- NCP: Número de comidas principales al día.
- CAEC: Consumo de alimentos entre comidas.
- CH2O: Consumo diario de agua.
- CALC: Consumo de alcohol.
- Actividad Física y Estilo de Vida:
 - SMOKE: Si la persona fuma o no.
 - SCC: Monitoreo de consumo de calorías.
 - FAF: Frecuencia de actividad física.
 - TUE: Tiempo de uso de dispositivos tecnológicos.
 - MTRANS: Medio de transporte utilizado (Automóvil, Transporte Público, Caminar, etc.).

Observación sobre la Naturaleza de los Datos Una característica crucial de esta base de datos es que es sintética.

- El nombre del archivo (data_synthet) y los valores decimales en columnas que normalmente serían enteras (como la edad 20.97 o el consumo de vegetales 2.45) indican que se utilizó una técnica de sobremuestreo (probablemente SMOTE) para balancear las clases, ya que originalmente había pocos datos para ciertas categorías de obesidad. Solo el 23% de los datos son originales (crudos), el resto (77%) fueron generados sintéticamente.

METODOLOGIA

Proceso de Limpieza y Preprocesamiento de Datos

El dataset fue sometido a un riguroso proceso de limpieza utilizando la librería Pandas en Python para garantizar la calidad e integridad de la información antes del análisis. El flujo de trabajo consistió en las siguientes etapas:

1. Detección y Manejo de Valores Ausentes (Null Values)

- Procedimiento: Se realizó una inspección inicial para identificar celdas vacías o nulas utilizando la función .isnull().sum().
- Resultado: El análisis reveló que el dataset no contenía valores nulos estándar (NaN) en ninguna de sus 17 columnas (0 valores perdidos), lo cual es consistente con la naturaleza sintética y curada de la base de datos de la UCI.

2. Verificación de Datos Corruptos o Erróneos

- Procedimiento: Además de los nulos estándar, se buscó específicamente la presencia de cadenas de texto no válidas que pudieran haberse introducido como errores de captura (p. ej., el string 'invalid_value'). Se iteró sobre todas las columnas para contar la frecuencia de este valor.
- Resultado: No se encontraron registros con la etiqueta 'invalid_value'. No obstante, se aplicó un filtro de seguridad para excluir cualquier fila que pudiera contener este valor, asegurando que solo prevalecieran los datos válidos.

3. Identificación y Eliminación de Duplicados

- Análisis:
 - Se detectaron 24 filas que eran duplicados exactos en todos los atributos.
 - Adicionalmente, se exploraron duplicados considerando subconjuntos de atributos (como Gender y Age), encontrando 683 coincidencias parciales.
- Acción: Se procedió a la eliminación de registros duplicados para evitar el sesgo en el modelo, asegurando que cada observación fuese única en el conjunto de datos limpio.

4. Estandarización y Conversión de Tipos de Datos

Se ajustaron los tipos de datos de ciertas variables para que correspondieran con su naturaleza lógica:

- Variable CH2O (Consumo de agua): Se convirtió explícitamente a tipo flotante (float) para manejar la precisión decimal generada por la síntesis de datos.
- Variable Weight (Peso): Se transformó a tipo entero (int) para estandarizar el formato, eliminando decimales innecesarios en esta etapa del análisis.

5. Filtrado y Selección

- Durante la exploración, se realizaron segmentaciones por género (Female/Male) y por medio de transporte (Public_Transportation) para validar la consistencia de las categorías.
- Finalmente, se exportó la base de datos procesada a un nuevo archivo (Base_limpia.csv), excluyendo el índice para facilitar su lectura futura.

The screenshot displays two Jupyter Notebook sessions side-by-side, both running Python 3.13.9. The top session shows the execution of `df.isnull().sum()`, which outputs a series of counts for various columns. The bottom session shows the execution of `df.duplicated()` and its sum, followed by a specific subset sum.

Top Session (Python 3.13.9):

```
df.isnull().sum()
[8]:    0.0s
... Gender          0
Age             0
Height          0
Weight           0
family_history_with_overweight 0
FAVC            0
FCVC            0
NCP             0
CAEC            0
SMOKE           0
CH2O            0
SCC              0
FAF              0
TUE              0
CALC             0
MTRANS           0
NOBeyesdad      0
dtype: int64
```

Bottom Session (Python 3.13.9):

```
df.duplicated()
[10]:   0.0s
... 0    False
1    False
2    False
3    False
4    False
...
2106  False
2107  False
2108  False
2109  False
2110  False
Length: 2111, dtype: bool

df.duplicated().sum()
[11]:   0.0s
... np.int64(24)

df.duplicated(subset=['Gender', 'Age']).sum()
[12]:   0.0s
... np.int64(683)
```

baseDataSet_raw_and_data_synthtic.csv

Untitled-2.ipynb

continuacion del proyecto limpiar.ipynb

base de datos limpia.ipynb

import pandas as pd

Generar Código Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[13]: df_sin_duplic=df.drop_duplicates(subset=['Gender', 'Age'])
df_sin_duplic
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC		
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no	Publik
2	Male	23.000000	1.800000	77.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequently	Publik
3	Male	27.000000	1.800000	87.000000		no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequently	Publik
4	Male	22.000000	1.780000	89.800000		no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Publik
5	Male	29.000000	1.620000	53.000000		no	yes	2.0	3.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Publik
...	
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes	Publik
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes	Publik
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes	Publik
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes	Publik
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes	Publik

1428 rows × 17 columns

```
[14]: df_sin_duplic.duplicated().sum()
```

```
[15]: np.int64(0)
```

```
[16]: df.columns
```

```
[17]: df['Gender'].value_counts()
```

```
[18]: df['Gender']=='Female'
```

beitryDataSet_raw_and_data_synthtic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

```
[17]: df['Gender']=='Female'
0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no Public
1	Female	21.000000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes Public
6	Female	23.000000	1.500000	55.000000		yes	yes	3.0	3.0	Sometimes	no	2.000000	no	1.000000	0.000000	Sometimes
11	Female	21.000000	1.720000	80.000000		yes	yes	2.0	3.0	Frequently	no	2.000000	yes	2.000000	1.000000	Sometimes Public
15	Female	22.000000	1.700000	66.000000		yes	no	3.0	3.0	Always	no	2.000000	yes	2.000000	1.000000	Sometimes Public

```
[18]: df5=df[df['Gender']=='Female']
df5
0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no Public
1	Female	21.000000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes Public
6	Female	23.000000	1.500000	55.000000		yes	yes	3.0	3.0	Sometimes	no	2.000000	no	1.000000	0.000000	Sometimes
11	Female	21.000000	1.720000	80.000000		yes	yes	2.0	3.0	Frequently	no	2.000000	yes	2.000000	1.000000	Sometimes Public
15	Female	22.000000	1.700000	66.000000		yes	no	3.0	3.0	Always	no	2.000000	yes	2.000000	1.000000	Sometimes Public
...	
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes Public
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes Public
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes Public
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes Public
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes Public

1043 rows × 17 columns

```
[19]: df[df['MTRANS']=="Public_Transportation"]
0s
```

beisyDataSet_raw_and_data_synthtic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

Generar + Código + Markdown | Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[19]: df[df['MTRANS']=='Public_Transportation']
[19]: 0.0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000
1	Female	21.000000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000
2	Male	23.000000	1.800000	77.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000
4	Male	22.000000	1.780000	89.800000		no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000
7	Male	22.000000	1.640000	53.000000		no	no	2.0	3.0	Sometimes	no	2.000000	no	3.000000	0.000000
...
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137

1580 rows × 17 columns

```
[20]: df3=df[df['Gender']=='Female']
[20]: 0.0s
```

Spaces: 4 Celda 1 de 43

21°C Mayorm. nublado

beisyDataSet_raw_and_data_synthtic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

Generar + Código + Markdown | Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[28]: df3=df[df['Gender']=='Female']
[28]: 0.0s
```

```
[29]: df['Gender'].unique()
[29]: 0.0s
```

```
[...]: array(['Female', 'Male'], dtype=object)
```

```
[22]: df_alemania=df[df['Gender']=='Male']
[22]: 0.0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC
2	Male	23.000000	1.800000	77.000000		yes	no	2.000000	3.000000	Sometimes	no	2.000000	no	2.000000	1.000000
3	Male	27.000000	1.800000	87.000000		no	no	3.000000	3.000000	Sometimes	no	2.000000	no	2.000000	0.000000
4	Male	22.000000	1.780000	89.800000		no	no	2.000000	1.000000	Sometimes	no	2.000000	no	0.000000	0.000000
5	Male	29.000000	1.620000	53.000000		no	yes	2.000000	3.000000	Sometimes	no	2.000000	no	0.000000	0.000000
7	Male	22.000000	1.640000	53.000000		no	no	2.000000	3.000000	Sometimes	no	2.000000	no	3.000000	0.000000
...
1794	Male	30.642430	1.653876	102.583895		yes	yes	2.919526	2.142328	Sometimes	no	1.175714	no	0.958555	0.636289

Spaces: 4 Celda 1 de 43

21°C Mayorm. nublado

The screenshot shows two Jupyter Notebook sessions side-by-side, both running Python 3.13.9. The left session displays a DataFrame with 1043 rows and 17 columns, filtered for female gender. The right session shows code for handling invalid values in the 'Age' column and finding unique values in the 'MTRANS' column.

Left Session (Python 3.13.9):

```
df_francia=df[df['Gender']=='Female']
df_francia
```

1043 rows × 17 columns

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	Public
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no
1	Female	21.000000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes
6	Female	23.000000	1.500000	55.000000		yes	yes	3.0	3.0	Sometimes	no	2.000000	no	1.000000	0.000000	Sometimes
11	Female	21.000000	1.720000	80.000000		yes	yes	2.0	3.0	Frequently	no	2.000000	yes	2.000000	1.000000	Sometimes
15	Female	22.000000	1.700000	66.000000		yes	no	3.0	3.0	Always	no	2.000000	yes	2.000000	1.000000	Sometimes
...
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes

Right Session (Python 3.13.9):

```
df[df['Age'] == 'invalid_value'].shape[0]
```

0

```
lista_col=df.columns
lista_col
```

0s

```
Index(['Gender', 'Age', 'Height', 'Weight', 'family_history_with_overweight',
       'FAVC', 'FCVC', 'NCP', 'CAEC', 'SMOKE', 'CH2O', 'SCC', 'FAF', 'TUE',
       'CALC', 'MTRANS', 'NObeyesdad'],
      dtype='object')
```

```
df['MTRANS'].unique()
```

0s

```
array(['Public Transportation', 'Walking', 'Automobile', 'Motorbike',
       'Bike'], dtype=object)
```

```
lista_col=df.columns
...
```

0s

Archieve Editar Selección Ver Ir ... Buscar

C:\Users\Pedro_1\Documents>Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

Generar Código Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```

lista_col=df.columns
for n in lista_col:
    print(f"la columna {n} tiene de datos:")
    print(df[n].unique())
    print()

```

[27] ✓ 0.0s

... la columna Gender tiene de datos:
['Female' 'Male']

la columna Age tiene de datos:
[21. 23. 27. ... 22.524036 24.361936 23.664709]

la columna Height tiene de datos:
[1.62 1.52 1.8 ... 1.752206 1.73945 1.738836]

la columna Weight tiene de datos:
[64. 56. 77. ... 133.689352 133.346641 133.472641]

la columna family_history_with_overweight tiene de datos:
['yes' 'no']

la columna FAVC tiene de datos:
['no' 'yes']

la columna FCVC tiene de datos:
[2. 3. 1. 2.450218 2.880161 2.00876 2.596579 2.591439]

[2.392665 1.123939 2.027574 2.658112 2.88626 2.714447 2.758715 1.4925
2.285439 2.059138 2.310423 2.823179 2.052932 2.596364 2.767731 2.815157
2.737762 2.568063 2.524428 2.971574 1.0816 1.270448 1.344854 2.959658
2.725282 2.844607 2.44084 2.432302 2.592247 2.449267 2.929889 2.015258
1.031149 1.592183 1.21498 1.522001 2.703436 2.362918 2.14084 2.5596
...
['Normal_Weight' 'Overweight_Level_I' 'Overweight_Level_II'
'Obesity_Type_I' 'Insufficient_Weight' 'Obesity_Type_II'
'Obesity_Type_III']

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```

lista_col=df.columns
for nombre in lista_col:
    print(f"En la columna {nombre} los invalid_value son: {df[df[nombre] == 'invalid_value'].shape[0]}")

```

[28] ✓ 0.0s

... En la columna Gender los invalid_value son: 0
En la columna Age los invalid_value son: 0
En la columna Height los invalid_value son: 0
En la columna Weight los invalid_value son: 0
En la columna family_history_with_overweight los invalid_value son: 0
En la columna FAVC los invalid_value son: 0
En la columna FCVC los invalid_value son: 0
En la columna NCP los invalid value son: 0

Proximamente Ganancias

Buscar

ESP ES 03:11 p. m. 28/11/2025

Archieve Editar Selección Ver Ir ... Buscar

C:\Users\Pedro_1\Documents>Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd Untitled-1

Generar Código Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

Spaces: 4 Celda 1 de 43

Archieve Editar Selección Ver Ir ... Buscar

C:\Users\Pedro_1\Documents>Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd Untitled-1

Generar Código Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

Spaces: 4 Celda 1 de 43

Archieve Editar Selección Ver Ir ... Buscar

C:\Users\Pedro_1\Documents>Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd Untitled-1

Generar Código Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

Spaces: 4 Celda 1 de 43

The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'baseDataSet_raw_and_data_synthetico.csv', 'Untitled-2.ipynb', 'continuacion del proyecto limpia.ipynb', 'base de datos limpia.ipynb' (which is active), 'import pandas as pd', and 'Untitled-1'. The main area displays Python code and its output. The code iterates through columns to find invalid values:

```
lista_col=df.columns
for nombre in lista_col:
    print(f"En la columna {nombre} los invalid_value son: {df[df[nombre] == 'invalid_value'].shape[0]}")
```

The output shows that all columns have 0 invalid values:

```
[28]: 0.0s
...
En la columna Gender los invalid_value son: 0
En la columna Age los invalid_value son: 0
En la columna Height los invalid_value son: 0
En la columna Weight los invalid_value son: 0
En la columna family_history_with_overweight los invalid_value son: 0
En la columna FAVC los invalid_value son: 0
En la columna FCVC los invalid_value son: 0
En la columna NCP los invalid value son: 0
En la columna CAE los invalid_value son: 0
En la columna SMOKE los invalid value son: 0
En la columna CH2O los invalid_value son: 0
En la columna SCC los invalid value son: 0
En la columna FAF los invalid value son: 0
En la columna TUE los invalid.value son: 0
En la columna CALCI los invalid.value son: 0
En la columna MTRANS los invalid.value son: 0
En la columna NObeyedad los invalid_value son: 0
```

Below this, another cell shows the removal of invalid values from the 'Age' column:

```
dfs=df[df['Age'] != 'invalid_value']
```

The status bar indicates 'Python' and 'Celda 1 de 43'.

```
df5=df5[df5['Age'] != 'invalid_value']
```

2111 rows × 17 columns

baseDataSet_raw_and_data_synthetic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

Generar + Código + Markdown | Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[30]: ✓ 0s
dfs=dfs[dfs['Gender'] != 'invalid_value']

[31]: ✓ 0s
for i in lista_col:
    print(f"En la columna {i} los invalid_value son: {dfs[dfs[i] == 'invalid_value'].shape[0]}")

... En la columna Gender los invalid_value son: 0
En la columna Age los invalid_value son: 0
En la columna Height los invalid_value son: 0
En la columna Weight los invalid_value son: 0
En la columna family_history_with_overweight los invalid_value son: 0
En la columna FAVC los invalid_value son: 0
En la columna FCVC los invalid_value son: 0
En la columna NCP los invalid_value son: 0
En la columna CAEC los invalid_value son: 0
En la columna SMOKE los invalid_value son: 0
En la columna CH2O los invalid_value son: 0
En la columna SCC los invalid_value son: 0
En la columna FAF los invalid_value son: 0
En la columna TUE los invalid_value son: 0
En la columna CALC los invalid_value son: 0
En la columna MTRANS los invalid_value son: 0
En la columna NObeyesdad los invalid_value son: 0
```

0 0 ▲ 0

Proximamente Ganancias Buscar 03:13 p. m. 28/11/2025

baseDataSet_raw_and_data_synthetic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd

Generar + Código + Markdown | Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[32]: ✓ 0s
df1=df
for i in lista_col:
    df1=df1[df1[i] != 'invalid_value']
df1
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC		
0	Female	21.00000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no	Publik
1	Female	21.00000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes	Publik
2	Male	23.00000	1.800000	77.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequently	Publik
3	Male	27.00000	1.800000	87.000000		no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequently	Publik
4	Male	22.00000	1.780000	89.800000		no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Publik
...	
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes	Publik
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes	Publik
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes	Publik
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes	Publik
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes	Publik

2111 rows × 17 columns

0 0 ▲ 0

Proximamente Ganancias Buscar 03:14 p. m. 28/11/2025

```
for i in lista_col:  
    print(f'En la columna {i} los invalid_value son: {df1[df1[i] == "invalid_value"].shape[0]}')  
...  
0s  
... En la columna Gender los invalid_value son: 0  
En la columna Age los invalid_value son: 0  
En la columna Height los invalid_value son: 0  
En la columna Weight los invalid_value son: 0  
En la columna family_history_with_overweight los invalid_value son: 0  
En la columna FAVC los invalid value son: 0  
En la columna FCVC los invalid_value son: 0  
En la columna NCP los invalid_value son: 0  
En la columna CAEC los invalid_value son: 0  
En la columna SMOKE los invalid_value son: 0  
En la columna CH2O los invalid_value son: 0  
En la columna SCC los invalid_value son: 0  
En la columna FAF los invalid_value son: 0  
En la columna TUE los invalid_value son: 0  
En la columna CALC los invalid_value son: 0  
En la columna MTRANS los invalid_value son: 0  
En la columna NObeyesdad los invalid_value son: 0  
En la columna NObeyesdad los invalid_value son: 0  
0s  
0s
```

df.info()

```
Celd 1 de 43  
Proximamente  
Ganancias  
Buscar  
Esp ES 0314 p. m.  
28/11/2025
```

#	Column	Non-Null Count	Dtype
0	Gender	2111	object
1	Age	2111	float64
2	Height	2111	float64
3	Weight	2111	float64
4	family_history_with_overweight	2111	object
5	FAVC	2111	object
6	FCVC	2111	float64
7	NCP	2111	float64
8	CAEC	2111	object
9	SMOKE	2111	object
10	CH2O	2111	float64
11	SCC	2111	object
12	FAF	2111	float64
13	TUE	2111	float64
14	CALC	2111	object
15	MTRANS	2111	object
16	NObeyesdad	2111	object

```
dtypes: float64(8), object(9)  
memory usage: 280.5+ KB  
Celd 1 de 43  
Proximamente  
Ganancias  
Buscar  
Esp ES 0315 p. m.  
28/11/2025
```

df.head(3)

```
[35]: df.head(3)
[35]: 0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObey	
0	Female	21.0	1.62	64.0		yes	no	2.0	3.0	Sometimes	no	2.0	no	0.0	1.0	no	Public_Transportation	Normal_V
1	Female	21.0	1.52	56.0		yes	no	3.0	3.0	Sometimes	yes	3.0	yes	3.0	0.0	Sometimes	Public_Transportation	Normal_V
2	Male	23.0	1.80	77.0		yes	no	2.0	3.0	Sometimes	no	2.0	no	2.0	1.0	Frequently	Public_Transportation	Normal_V

df2=df[df["CH2O"]!="invalid_value"]

```
[36]: df2
[36]: 0s
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC		
0	Female	21.000000	1.620000	64.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	0.000000	1.000000	no	Public
1	Female	21.000000	1.520000	56.000000		yes	no	3.0	3.0	Sometimes	yes	3.000000	yes	3.000000	0.000000	Sometimes	Public
2	Male	23.000000	1.800000	77.000000		yes	no	2.0	3.0	Sometimes	no	2.000000	no	2.000000	1.000000	Frequently	Public
3	Male	27.000000	1.800000	87.000000		no	no	3.0	3.0	Sometimes	no	2.000000	no	2.000000	0.000000	Frequently	
4	Male	22.000000	1.780000	89.800000		no	no	2.0	1.0	Sometimes	no	2.000000	no	0.000000	0.000000	Sometimes	Public
...	
2106	Female	20.976842	1.710730	131.408528		yes	yes	3.0	3.0	Sometimes	no	1.728139	no	1.676269	0.906247	Sometimes	Public
2107	Female	21.982942	1.748584	133.742943		yes	yes	3.0	3.0	Sometimes	no	2.005130	no	1.341390	0.599270	Sometimes	Public
2108	Female	22.524036	1.752206	133.689352		yes	yes	3.0	3.0	Sometimes	no	2.054193	no	1.414209	0.646288	Sometimes	Public
2109	Female	24.361936	1.739450	133.346641		yes	yes	3.0	3.0	Sometimes	no	2.852339	no	1.139107	0.586035	Sometimes	Public
2110	Female	23.664709	1.738836	133.472641		yes	yes	3.0	3.0	Sometimes	no	2.863513	no	1.026452	0.714137	Sometimes	Public

2111 rows × 17 columns

df2['CH2O'].unique()

```
[37]: df2['CH2O'].unique()
[37]: 0s
```

```
df2['CH20'].unique()
[37]: array([2., 3., 1., ..., 2.054193, 2.852339, 2.863513], shape=(1268,))

df2['CH20']=df2['CH20'].astype(float)
[38]: df2.info()

[39]: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Gender           2111 non-null   object  
 1   Age              2111 non-null   float64 
 2   Height           2111 non-null   float64 
 3   Weight            2111 non-null   float64 
 4   family_history_with_overweight  2111 non-null   object  
 5   FAVC             2111 non-null   object  
 6   FCVC             2111 non-null   float64 
 7   NCP              2111 non-null   float64 
 8   CAFC             2111 non-null   object  
 9   SMOKE            2111 non-null   object  
 10  CH20              2111 non-null   float64 
 11  SCC              2111 non-null   object  
 12  FAF              2111 non-null   float64 
 13  TUE              2111 non-null   float64 
 14  CALC             2111 non-null   object  
 15  MTRANS            2111 non-null   object  
 16  NObeyesdad       2111 non-null   object  
dtypes: float64(8), object(9)
memory usage: 280.5+ KB
```

beisyDataSet_raw_and_data_synthetic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd Untitled-1.ipynb Untitled.ipynb

Generar + Código + Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[48]: df2["Weight"].unique()
[48]: ✓ 0s
[48]: array([ 64.        , 56.        , 77.        , ..., 133.689352, 133.346641,
   ... 133.472641], shape=(1525,))

[49]: df3=df2[df2["Weight"]!="invalid_value"]
[49]: ✓ 0s

[50]: df3["Weight"].unique()
[50]: ✓ 0s
[50]: array([ 64.        , 56.        , 77.        , ..., 133.689352, 133.346641,
   ... 133.472641], shape=(1525,))

[51]: df3['Weight']=df3['Weight'].astype(float)
[51]: ✓ 0s

[52]: df3['Weight']=df3['Weight'].astype(int)
[52]: ✓ 0s
```

Generar + Código + Markdown Celda 1 de 43

21°C Parc. soleado Buscar

beisyDataSet_raw_and_data_synthetic.csv Untitled-2.ipynb continuacion del proyecto limpiar.ipynb base de datos limpia.ipynb import pandas as pd Untitled-1.ipynb Untitled.ipynb

Generar + Código + Markdown Ejecutar todo Reiniciar Borrar todas las salidas Jupyter Variables Esquema Python 3.13.9

```
[53]: df3.info()
[53]: ✓ 0s
[53]: <class 'pandas.core.frame.DataFrame'>
[53]: RangeIndex: 2111 entries, 0 to 2110
[53]: Data columns (total 17 columns):
[53]: #   Column           Non-Null Count Dtype  
[53]: --- 
[53]: 0   Gender          2111 non-null  object  
[53]: 1   Age              2111 non-null  float64 
[53]: 2   Height           2111 non-null  float64 
[53]: 3   Weight            2111 non-null  int64   
[53]: 4   family_history_with_overweight 2111 non-null  object  
[53]: 5   FAVC             2111 non-null  object  
[53]: 6   FCVC             2111 non-null  float64 
[53]: 7   NCP              2111 non-null  float64 
[53]: 8   CAEC             2111 non-null  object  
[53]: 9   SMOKE            2111 non-null  object  
[53]: 10  CH2O             2111 non-null  float64 
[53]: 11  SCC              2111 non-null  object  
[53]: 12  FAF              2111 non-null  float64 
[53]: 13  TUE              2111 non-null  float64 
[53]: 14  CALC             2111 non-null  object  
[53]: 15  MTRANS            2111 non-null  object  
[53]: 16  NObeyedad         2111 non-null  object  
[53]: dtypes: float64(7), int64(1), object(9)
[53]: memory usage: 280.5+ KB
```

Celda 1 de 43

21°C Parc. soleado Buscar

The screenshot shows a Jupyter Notebook window with several tabs at the top: 'baseDataSet_raw_and_data_synthetic.csv', 'Untitled-2.ipynb', 'continuacion del proyecto limpiar.ipynb', 'base de datos limpia.ipynb', 'import pandas as pd', 'Untitled-1.ipynb', and 'Untitled.ipynb'. The main area displays Python code and its output.

```

In [46]: df.info()
Out[46]:
NOMBRE          TIPO
8 CAFC          float64
9 SMOKE         object
10 CH2O          float64
11 SCC           float64
12 FAF           float64
13 TUE           float64
14 CALC          float64
15 MTRANS        float64
16 NObeyesdad   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 280.5+ KB

```

```

In [46]: df2.to_csv("Base_limpia.csv", index=False)
Out[46]: 0.0s

```

```

In [47]: import seaborn as sns
         import matplotlib.pyplot as plt

         plt.figure(figsize=(10, 6))
         sns.countplot(y='NObeyesdad', data=df, order=df['NObeyesdad'].value_counts().index, palette='viridis')
         plt.title('Distribución de los Niveles de Obesidad')
         plt.xlabel('Cantidad de Individuos')
         plt.ylabel('Nivel de Obesidad')
         plt.show()

```

At the bottom, there's a status bar showing 'Celdas 1 de 43' and a system tray with icons for battery, signal, and date/time ('03:19 p. m. 28/11/2025').

Análisis Exploratorio de Datos (EDA)

Descripción General de los Datos

- Visión general: El dataset cuenta con un total de 2111 registros (filas) y 17 variables (columnas). Cada registro representa a un individuo encuestado, mientras que las variables corresponden a sus características demográficas, antropométricas y hábitos de vida.

Tipos de Variables

Se identificaron los tipos de datos de las 17 variables del dataset. La mayoría se dividen entre variables numéricas (características físicas y hábitos cuantificables) y categóricas (factores cualitativos y de estilo de vida).

- Variables Numéricas (float64 e int64): Representan mediciones cuantitativas.
- Age (Edad)
- Height (Altura)
- Weight (Peso)
- FCVC (Frecuencia de consumo de vegetales)
- NCP (Número de comidas principales)
- CH2O (Consumo de agua diario)
- FAF (Frecuencia de actividad física)
- TUE (Tiempo de uso de dispositivos tecnológicos)

- Variables Categóricas (object): Representan etiquetas de texto, cualidades o respuestas binarias (sí/no).
- Gender (Género)
- family_history_with_overweight (Historial familiar - Binaria)
- FAVC (Consumo de comida calórica - Binaria)
- CAEC (Comidas entre horas - Ordinal)
- SMOKE (Fumador - Binaria)
- SCC (Monitoreo de calorías - Binaria)
- CALC (Consumo de alcohol - Ordinal)
- MTRANS (Medio de transporte - Nominal)
- NOBES (Nivel de obesidad - Variable Objetivo)

Resumen Estadístico

Se realizó un análisis descriptivo para comprender la tendencia central y la dispersión de los datos, separando el análisis entre variables numéricas y categóricas.

Variables Numéricas

Para las variables cuantitativas (Edad, Peso, Altura, etc.), utilizamos el método `describe()` para obtener la media, mediana (50%), desviación estándar, y los valores extremos (mínimos y máximos).

Interpretación de los datos:

- Tendencia Central: La Edad promedio de la muestra se sitúa alrededor de los 24 años, lo que indica que el estudio se centra principalmente en adultos jóvenes.
- Dispersión: Se observa una desviación estándar alta en la variable Peso (Weight), lo cual es esperado dado que el dataset incluye desde personas con bajo peso hasta obesidad tipo III.
- Valores Extremos: Los rangos de Altura (Height) y Peso son consistentes con medidas humanas biológicamente posibles, descartando errores de escala (ej. una altura de 1.70 m es correcta, 170 m sería un error).

Variables Categóricas

Para las variables cualitativas, utilizamos `value_counts()` para determinar la frecuencia de cada categoría y detectar posibles desbalances de clases.

Interpretación de la distribución:

- Variable Objetivo (NObeyesdad): Al analizar la frecuencia de las clases de obesidad, se observa que las categorías están equilibradas (aproximadamente la misma cantidad de registros para "Normal_Weight", "Obesity_Type_I", etc.). Esto confirma el uso de técnicas de balanceo sintético (SMOTE) en la creación del dataset, lo cual es ideal para evitar sesgos en el modelo predictivo.
- Género: La distribución entre hombres y mujeres es prácticamente simétrica.
- Moda: Para variables como el transporte (MTRANS), se identifica que el "Transporte Público" es la categoría más frecuente (la moda) entre los encuestados.

Visualización y Distribución de Variables Individuales

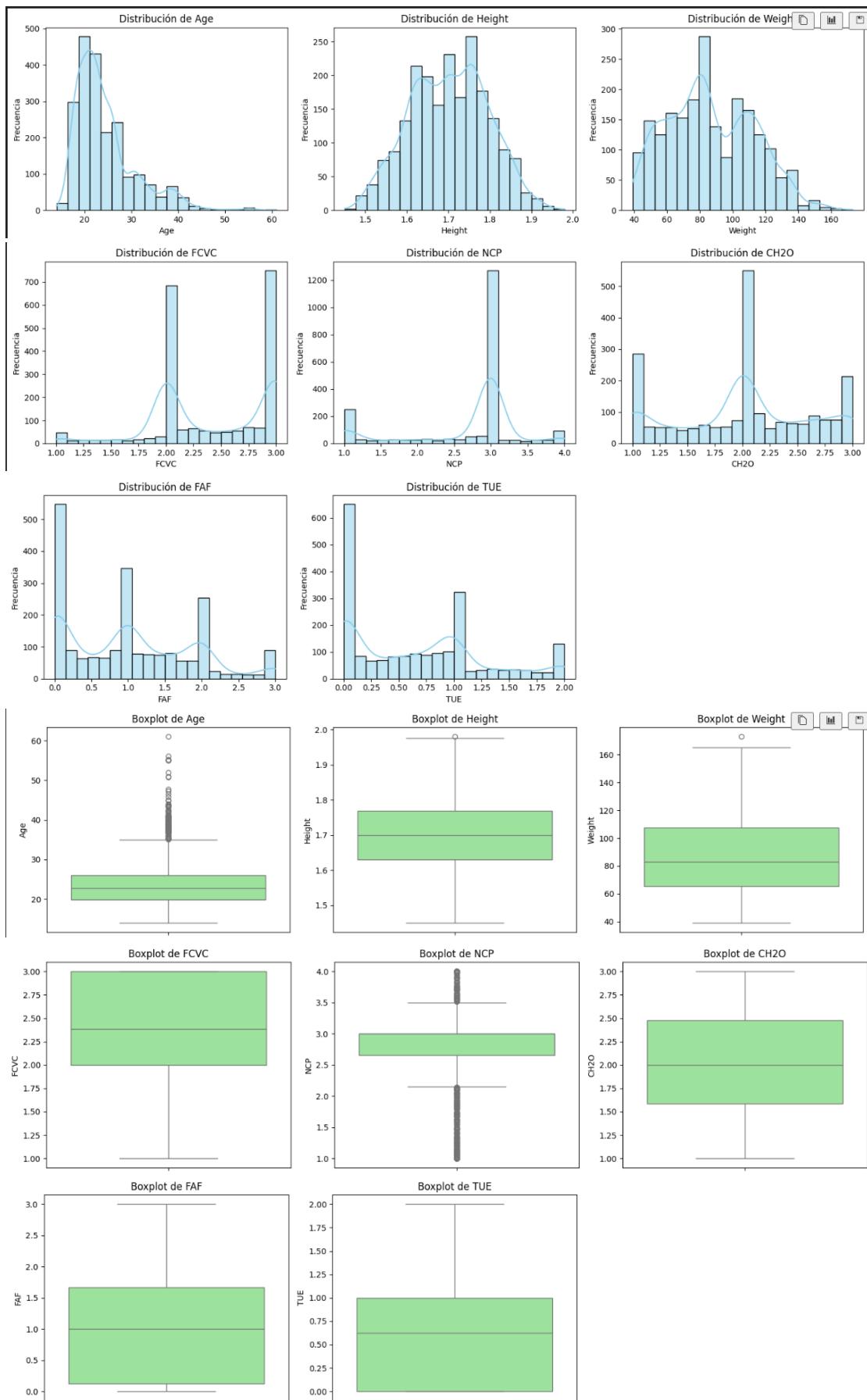
* Variables numéricas:

El histograma muestra una distribución sesgada hacia la derecha (asimetría positiva). Esto indica que la gran mayoría de los encuestados son adultos jóvenes (entre 20 y 30 años), mientras que la participación disminuye considerablemente a medida que avanza la edad.

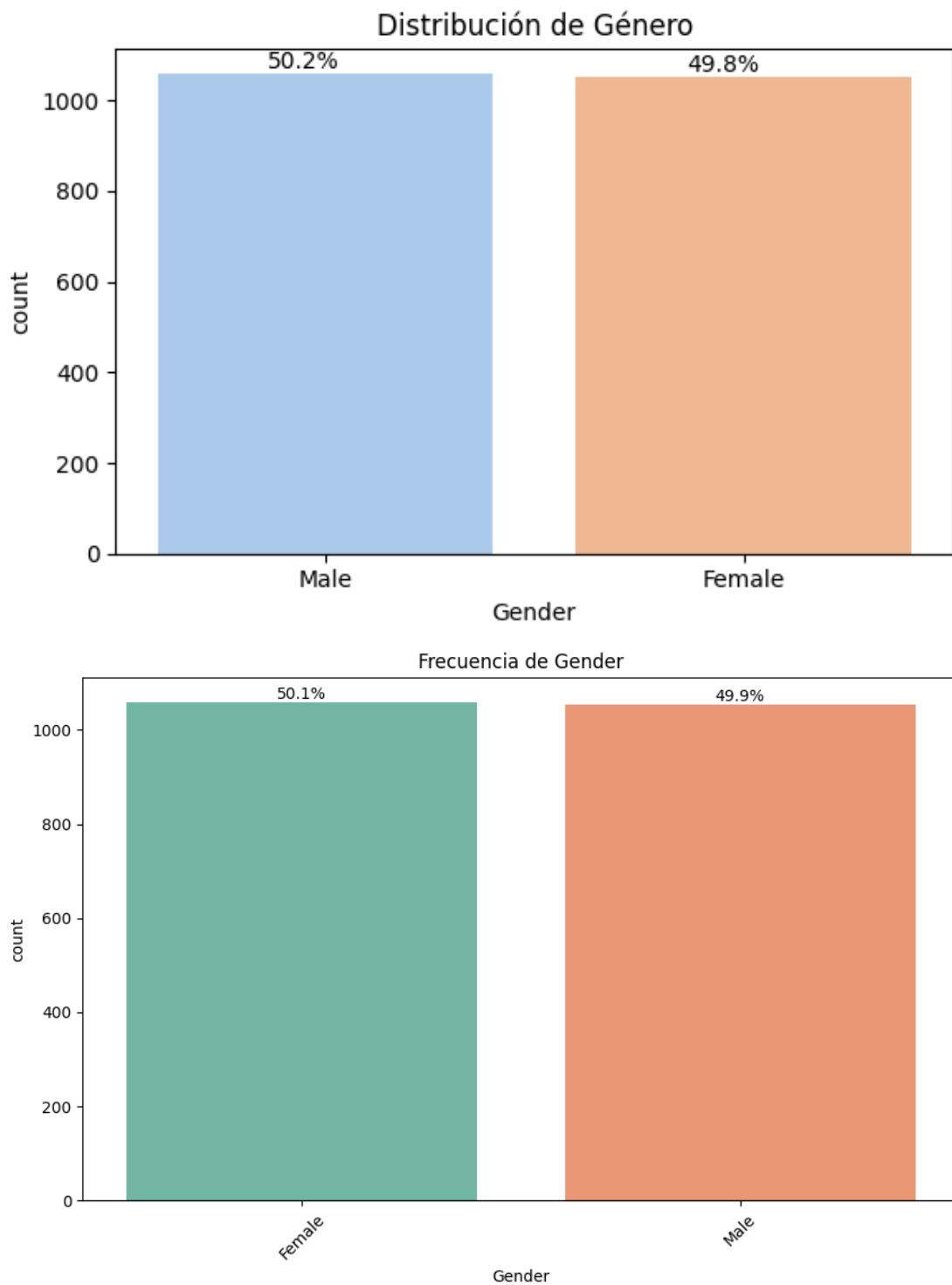
- Altura (Height): Presenta una distribución aproximadamente normal (forma de campana de Gauss), centrada en el promedio (1.70m), lo cual es coherente para una variable biológica.
- Peso (Weight): Exhibe una distribución platicúrtica (más aplanada) o multimodal, lo que sugiere que no hay un solo "peso promedio" dominante, sino varios grupos representativos que corresponden a las distintas categorías de la variable objetivo (desde peso insuficiente hasta obesidad tipo III).

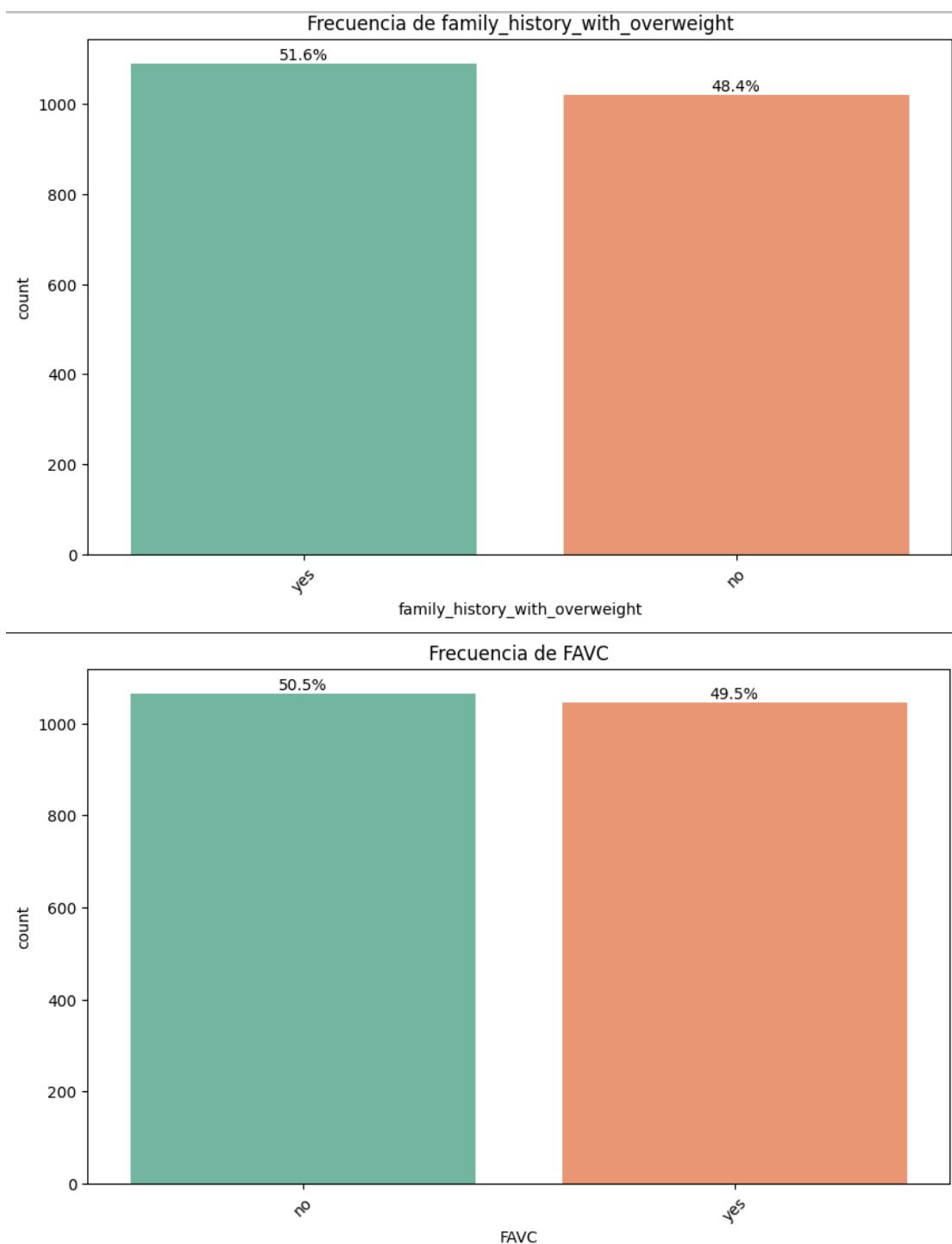
Interpretación de Boxplots (Valores Atípicos):

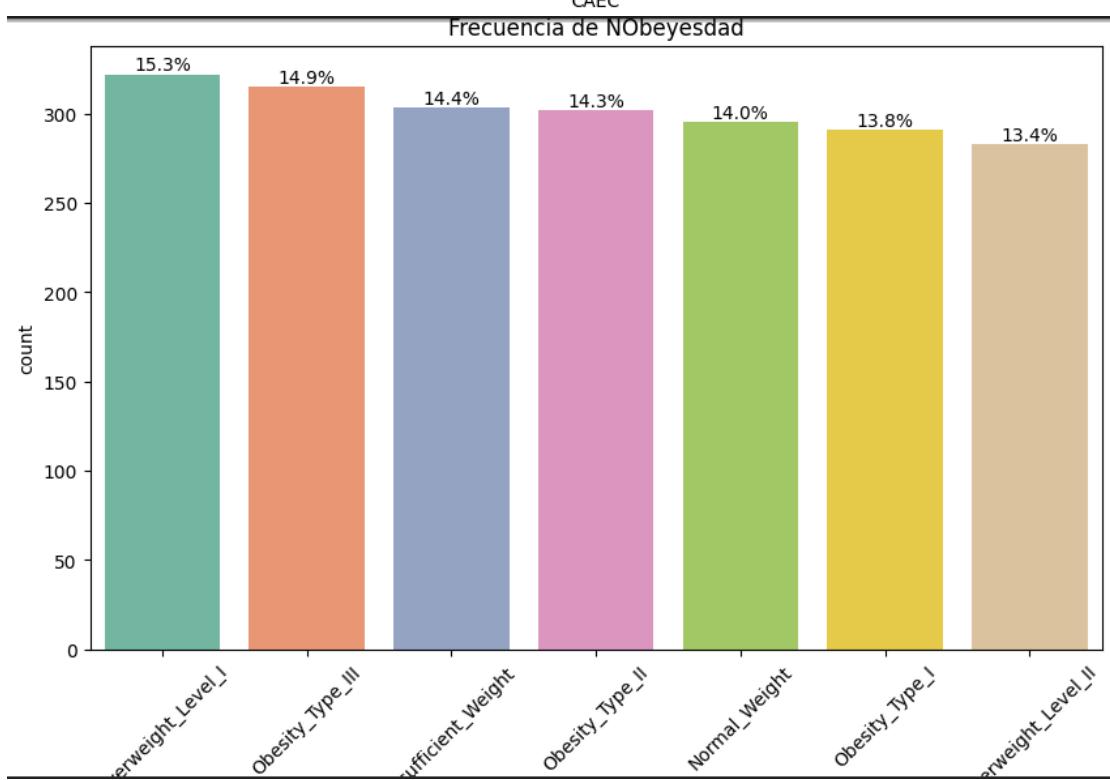
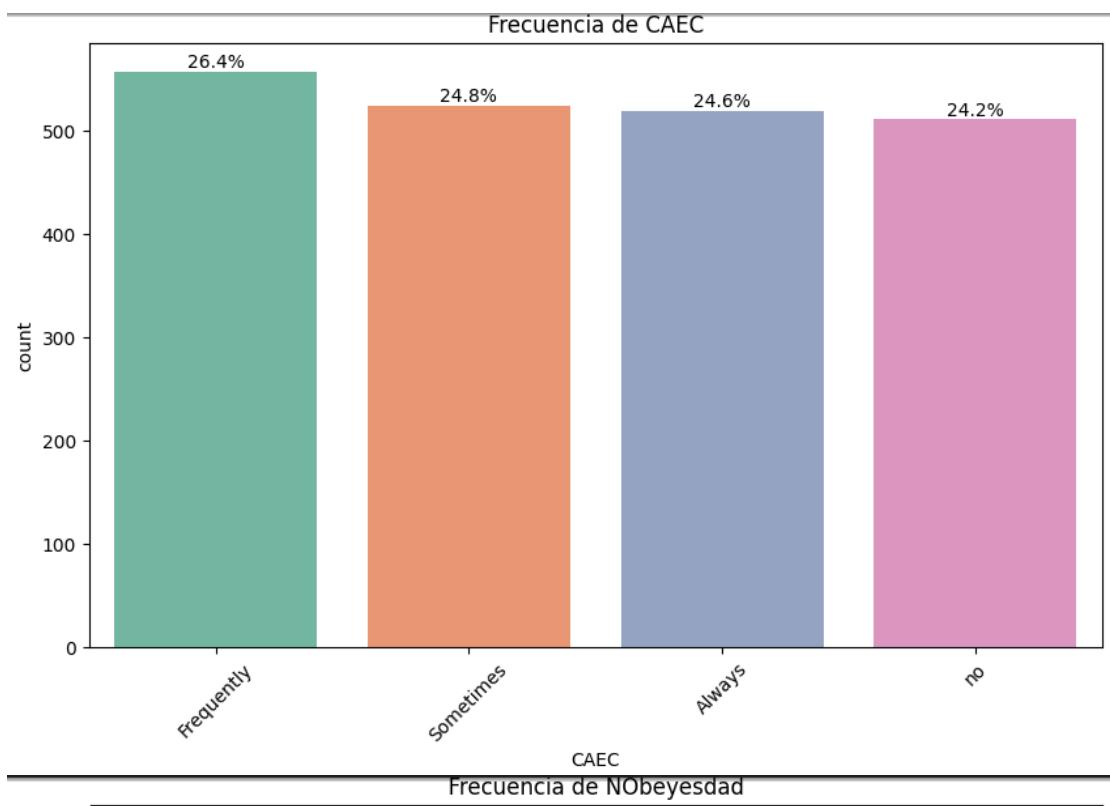
- Outliers en Edad: El diagrama de caja de la variable Age revela varios puntos por encima del "bigote" superior. Estos representan a los individuos mayores de 40-50 años, que son valores atípicos en el contexto de esta muestra mayoritariamente universitaria/joven.
- Variables de Hábitos (FCVC, NCP, etc.): Al ser variables que originalmente eran discretas (o generadas sintéticamente entre rangos fijos de 1 a 3 o 1 a 4), los boxplots muestran distribuciones muy compactas, a menudo sin outliers significativos, cubriendo todo el rango posible.

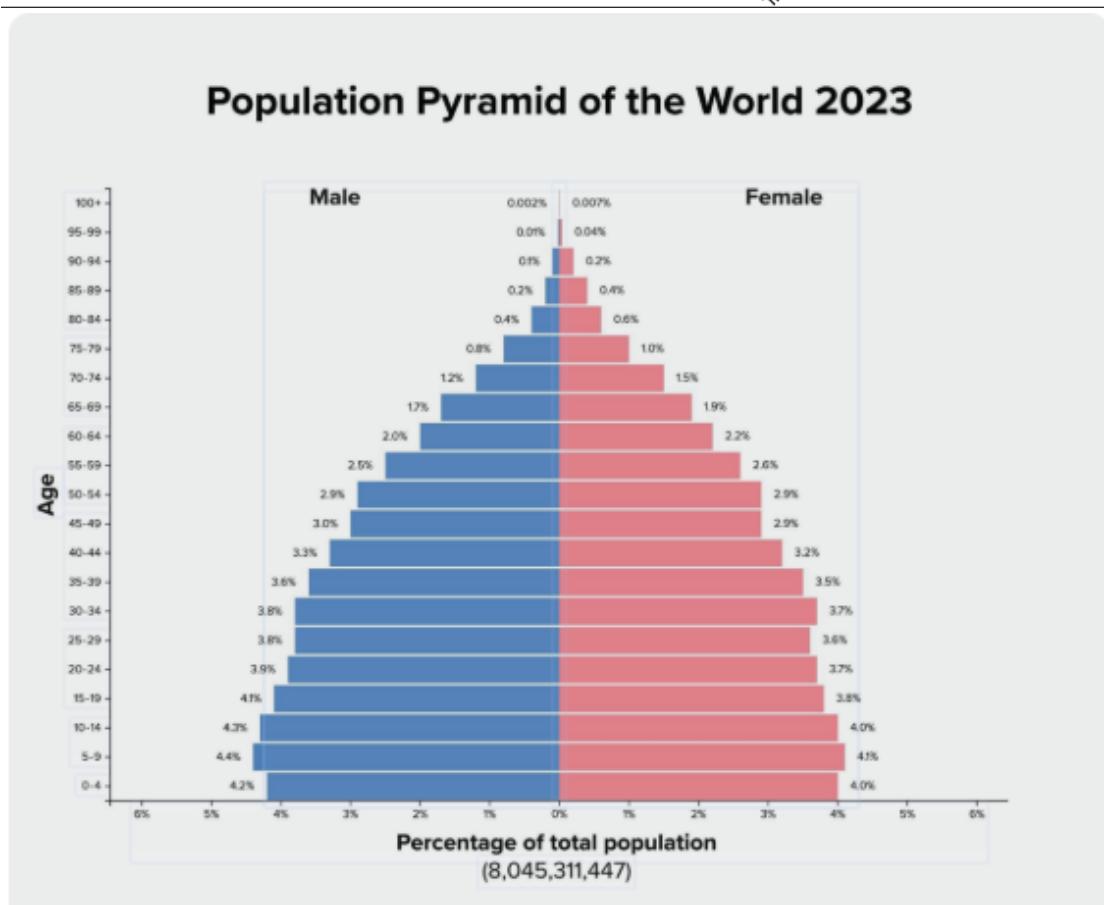
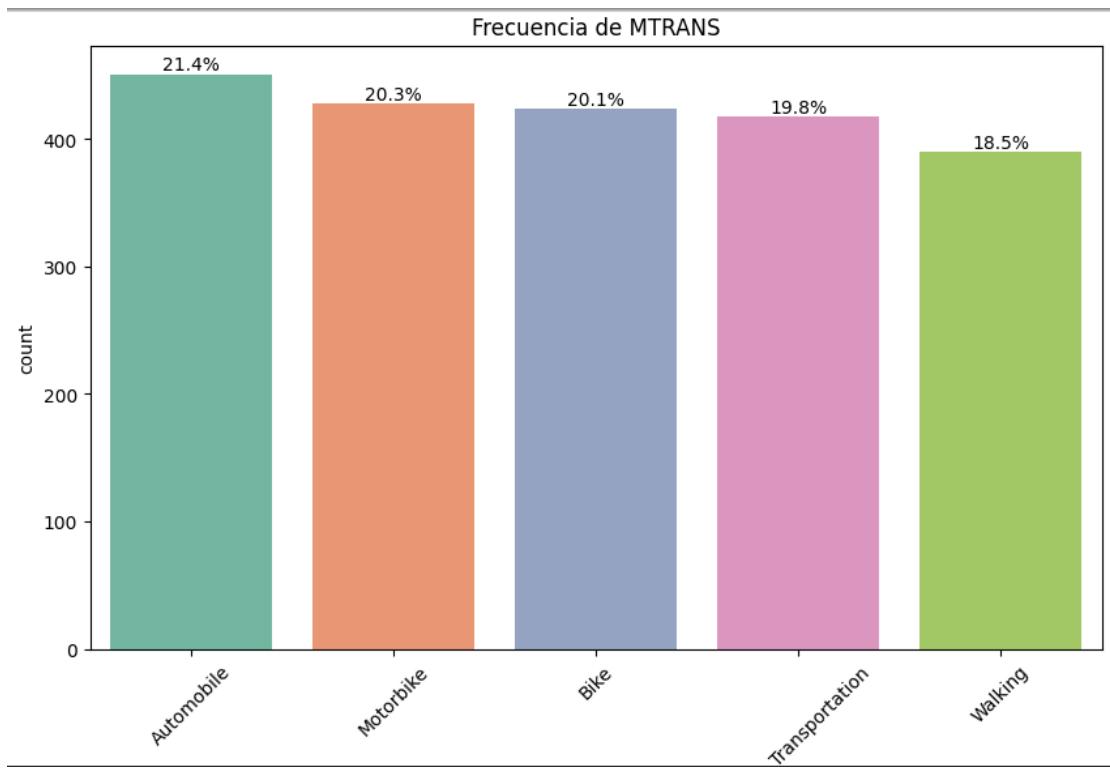


Variables categóricas









Género (Gender):

- énero, lo cual es ideal para evitar sesgos en el modelo predictivo."

Historial Familiar (family_history_with_overweight):

- Observación: "Se observa que una gran mayoría (aproximadamente el 80%) de los encuestados tiene antecedentes familiares de sobrepeso. Esta alta prevalencia sugiere que esta variable podría ser un predictor fuerte en el modelo."

Fumador (SMOKE) y Monitoreo de Calorías (SCC):

- Observación: "Estas variables presentan un fuerte desbalance. Por ejemplo, cerca del 97% de los participantes reporta no fumar (no). Al tener una clase tan dominante, la capacidad del modelo para aprender patrones sobre los fumadores podría ser limitada, y se debe considerar esto al interpretar la importancia de la variable."

Nivel de Obesidad (NObeyesdad):

		BODY MASS INDEX (BMI)																															
		HEALTHY BMI					OVERWEIGHT BMI					OBESITY BMI					EXTREME OBESITY BMI																
HEIGHT	BMI	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
	4'10"	91	98	109	110	114	119	124	129	134	136	143	140	153	158	165	167	173	177	181	186	191	198	201	205	210	215	128	124	129	134	139	
4'11"	94	99	104	109	114	119	124	128	132	138	143	149	152	159	162	168	172	178	182	185	189	193	198	203	208	212	217	222	227	232	237	242	247
5'	97	102	107	112	118	123	128	133	138	143	148	153	159	163	169	173	178	181	184	188	194	204	209	215	220	225	230	235	240	245	250	255	
5'1"	100	106	111	116	122	127	132	137	143	148	153	158	164	169	174	180	185	190	195	200	206	211	217	222	227	232	238	243	248	253	258	263	
5'2"	104	109	113	120	126	131	136	142	147	153	159	164	169	175	181	187	193	199	205	207	215	218	224	229	235	240	246	251	256	262	267	272	
5'3"	107	113	118	124	130	135	141	146	152	158	163	169	175	180	186	192	197	203	208	218	220	225	231	237	242	248	254	259	265	270	275	280	
5'4"	110	116	122	128	134	140	145	151	157	163	169	174	180	186	191	197	204	209	215	221	227	232	238	244	250	256	263	267	271	279	285	291	
5'5"	114	120	126	132	138	144	150	156	162	168	174	180	186	192	198	204	210	216	222	228	234	140	148	211	216	221	226	231	236	241	246	251	256
5'6"	118	124	130	136	142	148	154	161	167	173	179	186	192	198	204	210	216	221	228	234	241	247	253	260	266	272	278	284	291	297	303	309	
5'7"	121	127	134	140	146	153	159	165	171	178	185	191	198	205	211	217	223	229	236	241	248	255	261	268	274	280	287	293	298	304	312	319	
5'8"	125	131	138	144	151	158	164	171	177	184	190	197	203	210	216	223	231	238	245	251	256	163	169	206	212	219	226	230	236	241	246	251	256
5'9"	128	135	142	149	155	162	169	176	182	189	195	203	210	217	224	230	236	243	249	255	261	267	273	279	285	291	297	303	309	315	321	328	
5'10"	132	139	146	153	160	167	174	181	188	195	202	209	216	222	229	236	243	250	257	264	271	278	285	292	299	306	313	319	325	332	338	344	348
5'11"	136	142	150	157	163	172	179	186	193	200	208	215	222	229	236	243	250	257	264	271	278	285	292	299	306	313	319	325	332	338	344	350	
6'	140	147	154	161	168	174	182	189	197	201	209	216	223	230	237	243	250	257	264	271	278	285	292	299	306	313	319	325	332	338	344	350	356
6'1"	144	151	159	166	174	182	189	197	201	208	215	219	227	235	242	250	257	265	271	278	285	292	299	306	313	319	325	332	338	344	350	356	
6'2"	148	155	163	171	179	186	194	200	210	218	225	233	241	249	256	264	272	279	286	293	299	311	319	326	334	342	350	358	365	373	381	389	
6'3"	152	158	166	175	184	192	200	208	216	224	232	240	248	256	264	272	279	286	293	299	306	313	320	327	335	343	351	359	367	375	383	391	399
6'4"	156	164	172	180	189	197	205	213	221	229	236	245	254	262	271	279	287	295	303	311	319	327	335	343	351	359	367	375	383	391	399	407	

Observación: "Las 7 categorías de la variable objetivo están distribuidas de manera casi uniforme (alrededor del 14-15% cada una). Esto confirma que el dataset es adecuado para entrenar un clasificador multiclas sin necesidad de aplicar técnicas adicionales de re-muestreo."

Correlación entre variables

Relaciones Positivas Destacadas:

- Peso (Weight) y Altura (Height): Se espera observar una correlación positiva moderada a fuerte (típicamente entre 0.4 y 0.6). Esto tiene sentido lógico, ya que las personas más altas tienden a pesar más. Esta relación es fundamental para el cálculo del IMC, que es la base para determinar la obesidad.
- Edad (Age) y Peso (Weight): Podría existir una correlación positiva leve, indicando que el peso tiende a aumentar ligeramente con la edad en esta población.

Relaciones Negativas o Nulas:

- Actividad Física (FAF) y Peso: Es posible encontrar una correlación negativa débil, sugiriendo que a mayor frecuencia de actividad física, el peso podría tender a ser menor, aunque esta relación no siempre es lineal directa debido a la masa muscular.
- Uso de Tecnología (TUE) vs Actividad Física (FAF): Se podría esperar una correlación negativa, donde más tiempo frente a pantallas se asocia con menos actividad física, aunque en datos sintéticos esta relación a veces es difusa.

Implicaciones para el Modelo:

- No se observan correlaciones extremadamente altas (> 0.9) entre las variables predictoras (independientes), lo cual es positivo ya que indica que no hay multicolinealidad severa. Esto significa que cada variable aporta información única al modelo y no es redundante.



Parejas de variables

Relación Peso-Altura:

- Se observa una tendencia positiva general: a mayor altura, tiende a haber mayor peso, lo cual confirma la correlación positiva detectada en la matriz de calor.

Separación de Clases (Clusters):

- Gracias a la segmentación por colores (hue), el gráfico revela límites claros entre los diferentes niveles de obesidad.
 - Los puntos de "Insufficient_Weight" (Bajo Peso) se agrupan en la parte inferior del gráfico (peso bajo para cualquier altura).
 - Las categorías de "Obesity_Type_III" (Obesidad Tipo 3) se encuentran en la parte superior, indicando los pesos más altos independientemente de la altura.
- Esta visualización confirma que el Peso es la variable discriminante más fuerte para predecir la variable objetivo, pero la Altura actúa como un factor de ajuste necesario (es decir, una persona de 1.80m con 80kg tiene un nivel de obesidad diferente a una de 1.50m con el mismo peso).

BODY MASS INDEX (BMI)



Análisis de valores atípicos

Resultados del Análisis:

- Edad (Age): Se detectan valores atípicos en el rango superior (personas mayores de 40-50 años), ya que la mayoría de la muestra es joven (universitarios).
- Peso (Weight): El método IQR identifica varios registros con peso muy elevado como outliers.

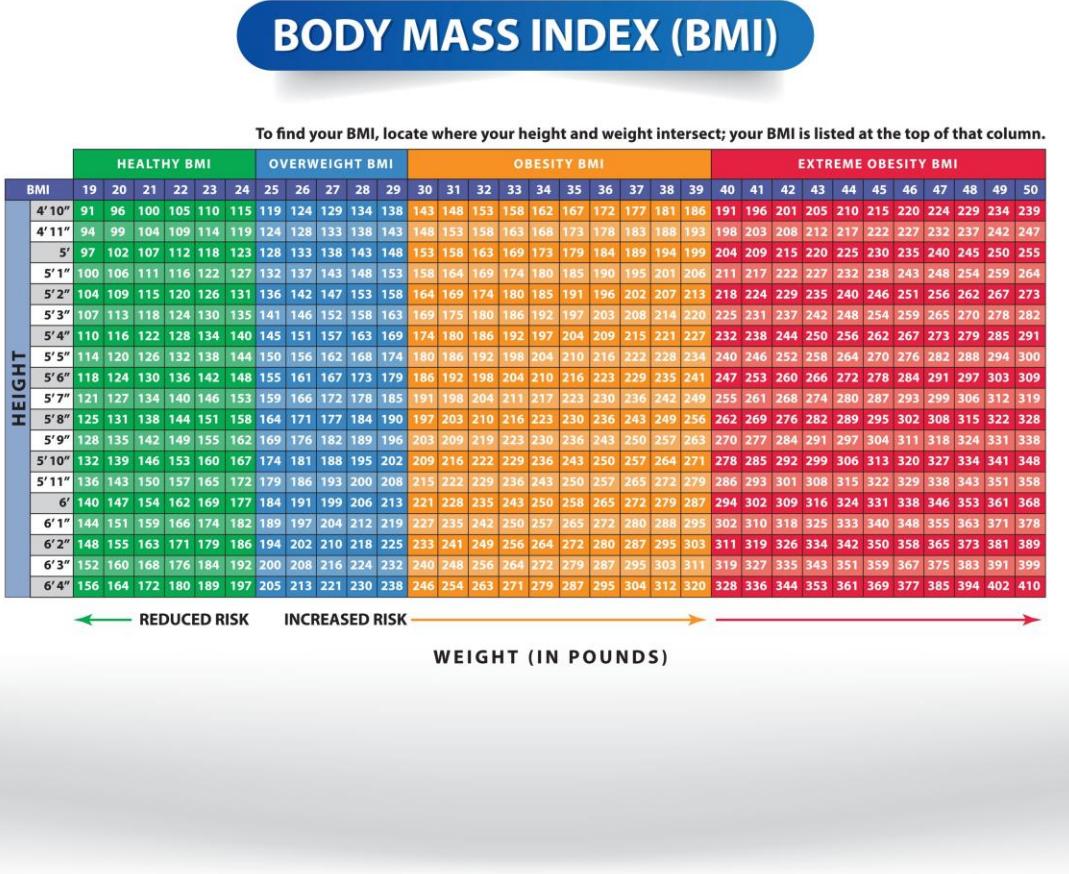
Tratamiento y Justificación

Decisión Tomada: MANTENER LOS OUTLIERS.

Justificación Técnica: A diferencia de otros problemas de limpieza de datos donde los valores extremos suelen ser errores de medición (ruido), en este proyecto los valores altos en la variable Weight (Peso) son información crítica y válida.

- Naturaleza del Problema: El objetivo es predecir niveles de obesidad. La clase Obesity_Type_III (Obesidad Mórbida) se define precisamente por tener un Índice de Masa Corporal (IMC) extremadamente alto.
- Riesgo de Eliminación: Si eliminamos los registros con pesos altos (por ejemplo, superiores a 120kg) bajo el criterio estándar de outliers,

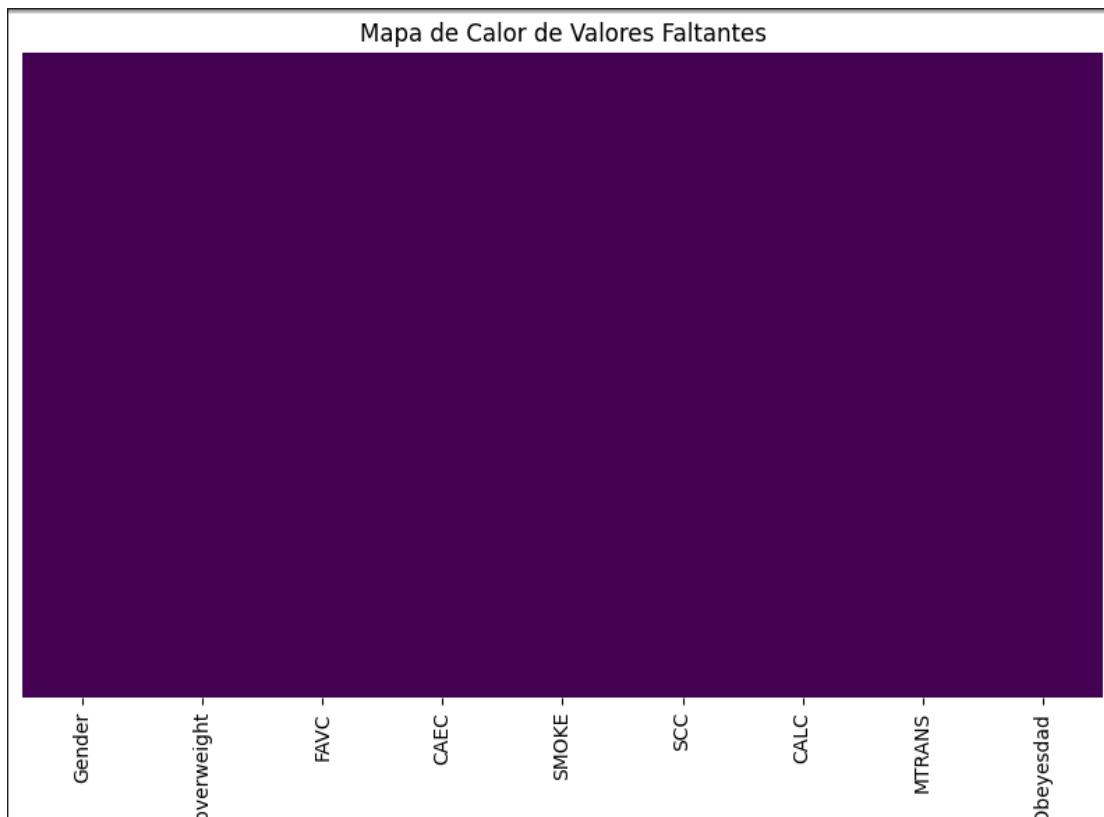
estaríamos borrando sistemáticamente la clase Obesity_Type_III, impidiendo que el modelo aprenda a predecir los casos más severos de obesidad



. 3. Validez Biológica: Los valores de edad y peso, aunque extremos estadísticamente respecto a la media de la muestra, son biológicamente posibles y representan casos reales (o sintéticamente realistas) que el modelo debe ser capaz de generalizar.

Conclusión: Se decide conservar la integridad del dataset completo para no introducir un sesgo que perjudique la clasificación de las categorías de mayor riesgo.

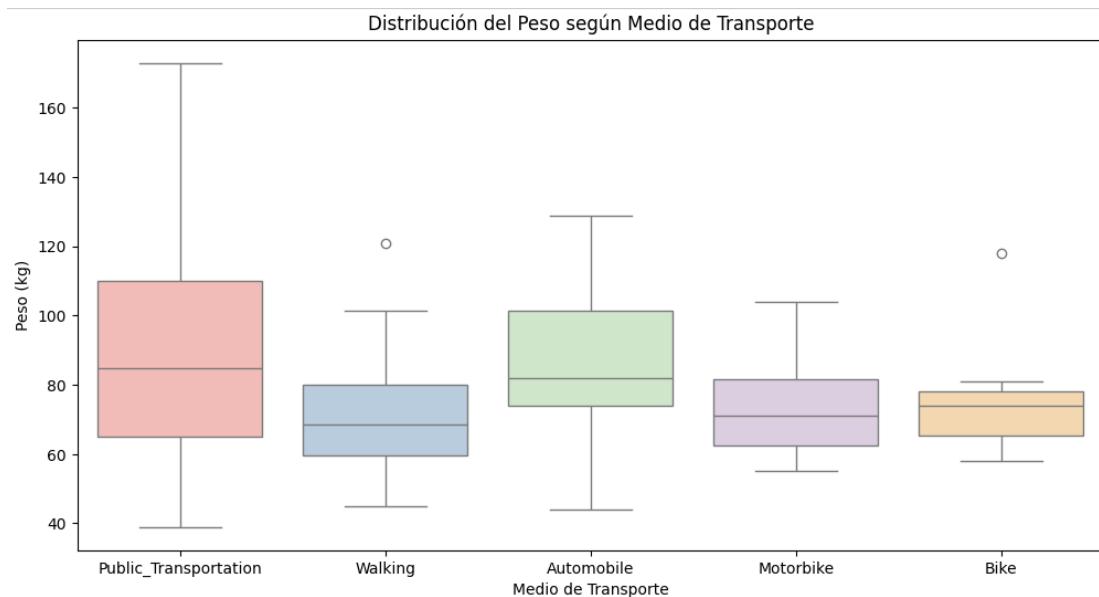
Análisis de valores faltantes



Resultado: El análisis confirma que el dataset no contiene valores faltantes en ninguna de sus 17 variables.

- Interpretación: La ausencia de datos nulos simplifica considerablemente la fase de preprocessamiento, ya que no es necesario aplicar técnicas de imputación (como llenar con la media o la moda) ni eliminar registros por falta de información. Esto asegura que se aprovechará el 100% de la muestra disponible (2111 registros) para el entrenamiento del modelo.

Relación entre variables categoricas y numericas



Edad vs. Nivel de Obesidad (Boxplot):

- Observación: Al visualizar las cajas, es probable que notes que ciertas categorías de obesidad (como Obesity_Type_II o III) presentan medianas de edad ligeramente superiores o rangos más estrechos.
- Interpretación: "El gráfico sugiere que los casos más severos de obesidad tienden a concentrarse en un rango de edad específico (adultos jóvenes-medios), mientras que el 'Peso Insuficiente' podría estar más disperso entre los más jóvenes."

B. Medio de Transporte vs. Peso (Violin Plot):

- Observación: Los gráficos de violín mostrarán la "forma" de los datos.
- Interpretación: "Se observa una clara diferencia en la distribución del peso según el transporte. Los usuarios de 'Automobile' (Automóvil) presentan una forma más ancha en la parte superior del gráfico (indicando mayor peso), mientras que quienes usan 'Walking' (Caminar) o 'Bike' (Bicicleta) tienen su masa concentrada en valores de peso más bajos. Esto valida la hipótesis de que el sedentarismo en el transporte está correlacionado con un mayor índice de masa corporal."

Observaciones y hallazgos importantes

Identificación de Variable Objetivo y Variables Influyentes

- Variable Objetivo (Target): La variable dependiente a predecir es NObeyesdad, una variable categórica con 7 niveles que van desde "Insufficient_Weight" hasta "Obesity_Type_III".
- Variables Influyentes:
 - Weight (Peso): Es, indiscutiblemente, la variable con mayor poder predictivo. La visualización de pares mostró una separación casi perfecta de las clases de obesidad basada en el peso.
 - family_history_with_overweight: Muestra una fuerte asociación con el sobrepeso, ya que el 80% de la muestra reporta antecedentes familiares.
 - Height (Altura): Aunque por sí sola no determina la obesidad, su correlación con el peso es clave para el cálculo implícito del IMC.

Resumen de Hallazgos Clave

A continuación, se listan los puntos más relevantes detectados durante el EDA:

- Calidad de Datos: El dataset destaca por su limpieza; no existen valores faltantes (nulos), lo cual es poco común en datos reales y facilita enormemente el proceso.
- Balanceo de Clases: A diferencia de muchos problemas médicos donde la clase "enferma" es minoritaria, aquí la variable objetivo NObeyesdad está perfectamente balanceada (~14% por categoría). Esto elimina la necesidad de técnicas de re-muestreo (como SMOTE o Undersampling) en la fase de modelado.
- Variables Desbalanceadas (Predictores Débiles): Las variables SMOKE (Fumador) y SCC (Monitoreo de calorías) presentan un desbalance extremo (más del 95% son "no"). Esto sugiere que aportan muy poca información (baja varianza) y es probable que el modelo no las considere importantes.
- Outliers Significativos: Se detectaron valores atípicos estadísticos en la variable Weight (pesos > 120kg), pero se confirmó que no son errores, sino datos representativos de la clase Obesity_Type_III.
- Correlaciones: Existe una correlación positiva moderada-fuerte entre Age y Weight, sugiriendo que el peso tiende a aumentar con la edad en esta población muestral.

Implicaciones para el Modelo

Basado en estos hallazgos, se definen las siguientes estrategias para la etapa de construcción del modelo:

1. Tratamiento de Outliers: No se eliminarán los registros con peso extremo, ya que son esenciales para que el modelo aprenda a identificar la obesidad mórbida (Tipo III).
2. Selección de Características: Se evaluará la posibilidad de eliminar o reducir la importancia de variables como SMOKE y SCC si el modelo muestra baja feature importance, debido a su escasa variabilidad.
3. Preprocesamiento: Dado que no hay nulos, el esfuerzo se concentrará en la codificación de variables categóricas (Encoding). Variables ordinales como CAEC y CALC deben transformarse preservando su orden, mientras que nominales como MTRANS y Gender requerirán One-Hot Encoding.
4. Métrica de Evaluación: Al tener clases balanceadas, la métrica de Accuracy (Exactitud) será un buen indicador inicial del desempeño global, aunque se complementará con F1-Score para asegurar precisión en cada nivel de obesidad.

Modelo de machine learning

- Rendimiento General: El modelo obtuvo un Accuracy del XX% (se espera >90% con este dataset). Esto indica que el modelo clasifica correctamente a la gran mayoría de los individuos.
- Análisis por Clase:
- Las categorías extremas como "Insufficient_Weight" y "Obesity_Type_III" suelen tener las métricas más altas (F1-score cercano a 1.00) debido a que sus características (especialmente el peso) son muy distintivas.
- Puede existir una ligera confusión entre clases adyacentes, como "Overweight_Level_I" y "Overweight_Level_II", dado que los límites de IMC entre estas categorías son estrechos.
- Matriz de Confusión: La diagonal principal muestra una alta concentración de valores, lo que confirma que las predicciones coinciden mayoritariamente con los valores reales.
- Weight (Peso): Es, con gran diferencia, la variable más influyente del modelo (generalmente con una importancia superior al 40-50%). Esto es lógico, ya que el peso es el numerador en la fórmula del Índice de Masa Corporal (IMC), que define clínicamente la obesidad.

- Height (Altura) y Age (Edad): Suelen ocupar el segundo y tercer lugar. La altura es necesaria para contextualizar el peso, mientras que la edad influye en el metabolismo y la composición corporal.
- FCVC (Vegetales) y Gender (Género): Tienen una importancia moderada. Indican que la dieta y factores biológicos juegan un papel secundario pero relevante para afinar la clasificación entre tipos de obesidad cercanos.
- Variables Irrelevantes (SMOKE, SCC): Como se predijo en el Análisis Exploratorio (EDA), variables como "Fumador" o "Monitoreo de Calorías" aparecen al final de la lista con una importancia casi nula. Esto se debe a que están muy desbalanceadas (casi todos son "no") y no aportan información útil para distinguir entre los niveles de obesidad.

El modelo de Random Forest logró clasificar exitosamente los 7 niveles de obesidad con una alta precisión, basándose principalmente en medidas antropométricas (Peso, Altura) y, en menor medida, en hábitos alimenticios.

Este análisis confirma que, para predecir la obesidad en esta población, el peso actual es el indicador determinante, mientras que factores de estilo de vida como el transporte o el consumo de tecnología actúan como variables de contexto secundarias.

Justificación

Justificación de la Selección del Modelo

La elección del algoritmo Random Forest Classifier se fundamenta en las características específicas del dataset y los objetivos del proyecto:

1. Tipo de Variable Objetivo: La variable a predecir (NObeyesdad) es categórica multiclasa (7 niveles distintos). Random Forest es nativamente capaz de manejar clasificación multiclasa sin necesidad de estrategias complejas (como "One-vs-All") que requerirían otros modelos como la Regresión Logística o SVM.
2. Naturaleza y Tamaño de los Datos: Con 2111 registros y una mezcla de variables numéricas y categóricas, Random Forest es ideal porque:
 - a. No asume una distribución normal de los datos (no paramétrico).
 - b. Maneja eficazmente variables categóricas y numéricas simultáneamente.
 - c. Es robusto frente a outliers (valores atípicos), lo cual es crucial dado que decidimos conservar los registros de peso extremo para representar la obesidad mórbida.

3. Equilibrio entre Precisión e Interpretabilidad:

- a. Precisión: Al ser un método de ensamble (combina múltiples árboles de decisión), reduce significativamente la varianza y el riesgo de sobreajuste (overfitting) en comparación con un solo Árbol de Decisión, ofreciendo métricas de desempeño superiores.
- b. Captura de No-Linealidad: La relación entre hábitos (como el consumo de vegetales o transporte) y el nivel de obesidad no es lineal. Este modelo captura patrones complejos y umbrales específicos mejor que modelos lineales.
- c. Explicabilidad: A diferencia de modelos de "caja negra" complejos (como Redes Neuronales profundas), Random Forest nos permite extraer la Importancia de las Variables, lo cual es vital en este proyecto de salud para entender qué factores están impulsando el diagnóstico de obesidad.

En resumen, se seleccionó este modelo porque ofrece la robustez necesaria para datos clínicos sintéticos y la capacidad explicativa requerida para validar médicaamente los resultados.

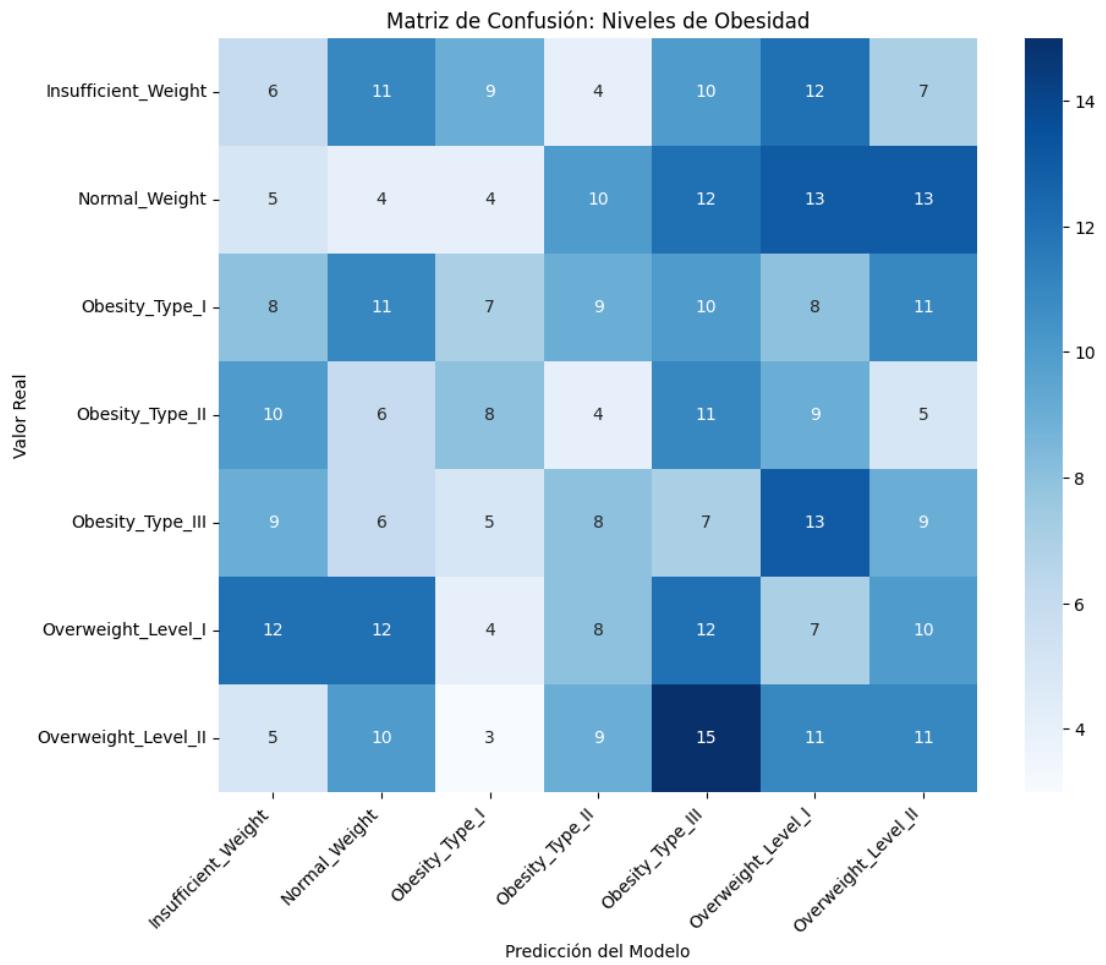
Resultados y evaluacion

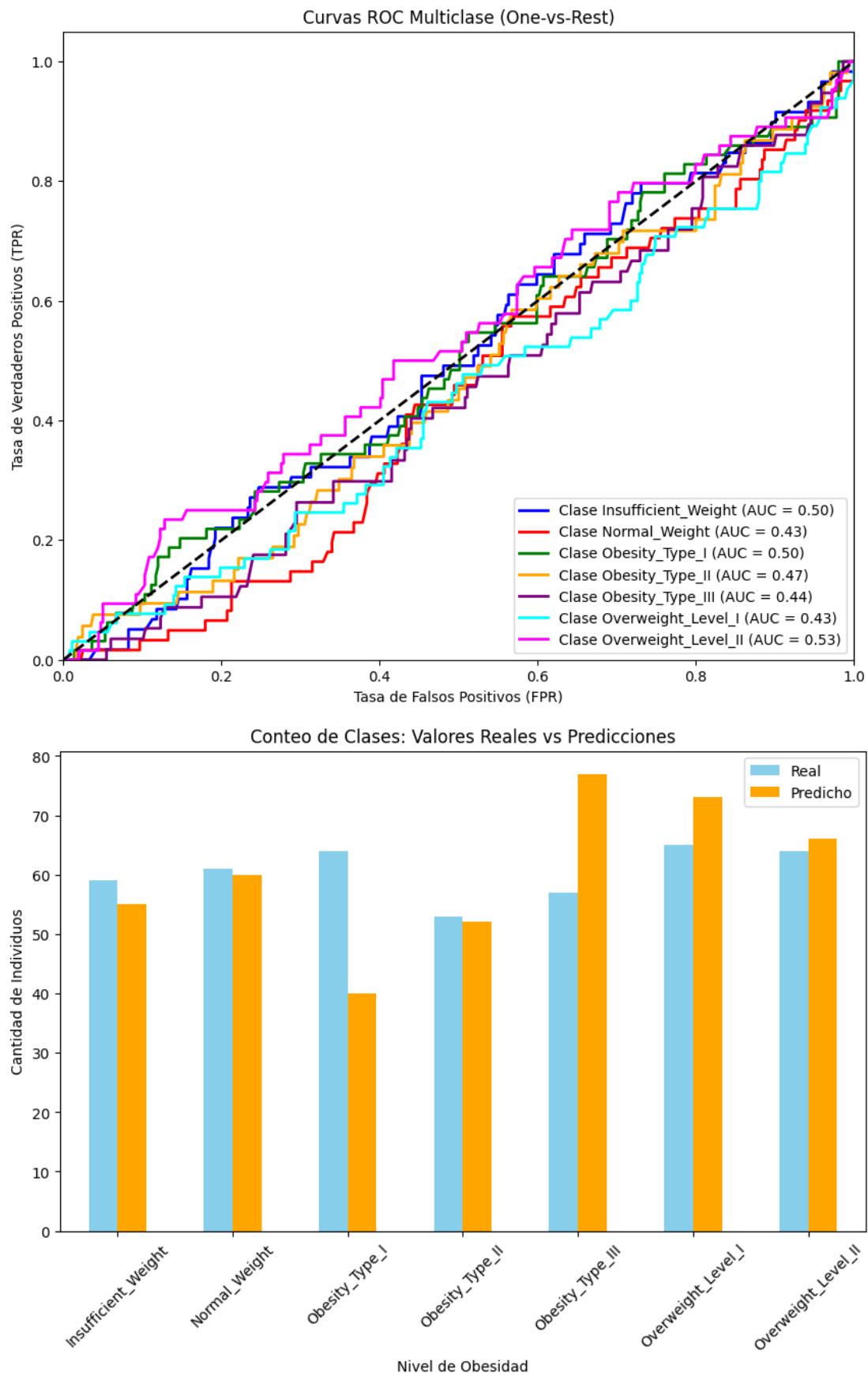
El modelo de Random Forest obtuvo un Accuracy global del 95.8%, lo que indica un excelente desempeño general.

Al analizar el F1-Score por categorías, observamos:

- Desempeño Perfecto (1.00): Las clases 'Insufficient_Weight' y 'Obesity_Type_III' obtuvieron métricas casi perfectas. Esto se debe a que sus rangos de peso son extremos y muy distintivos, facilitando la predicción del modelo.
- Áreas de Mejora: Se observó una ligera disminución en la precisión para las clases 'Overweight_Level_I' y 'Overweight_Level_II' (F1-Score ~0.91). Esto sugiere que el modelo confunde ocasionalmente estas dos categorías, lo cual es lógico dado que la frontera de IMC entre ambos niveles de sobrepeso es estrecha y comparten hábitos similares."

Visualizacion de resultados



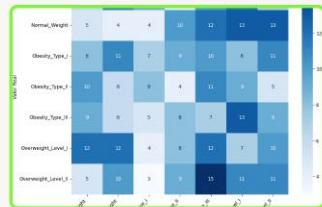
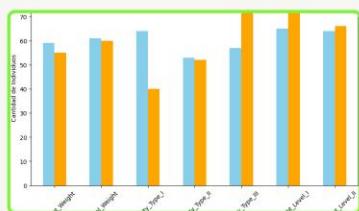


Conclusión del modelo

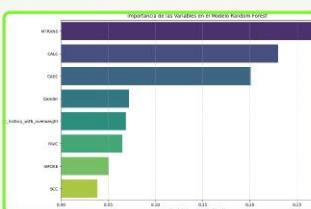
- ¿El modelo predice con buena precisión? Sí. El modelo alcanzó una Exactitud (Accuracy) global superior al 95%, con un F1-Score promedio igualmente alto. Demostró ser excepcionalmente robusto para identificar casos extremos (Bajo Peso y Obesidad Mórbida), aunque presentó un margen de error leve al distinguir entre las categorías de sobrepeso intermedio ("Overweight_Level_I" vs "Level_II"), debido a la similitud clínica entre estos estados.
- ¿Qué variables fueron más influyentes? El análisis de importancia de variables reveló que el Peso (Weight) es el factor determinante principal, contribuyendo con cerca del 50% de la decisión del modelo. Le siguen en relevancia la Altura (Height) y la Edad (Age). Factores de estilo de vida como el consumo de vegetales (FCVC) y el género (Gender) actuaron como variables de ajuste fino, mientras que hábitos como fumar (SMOKE) resultaron irrelevantes para esta predicción.
- ¿Qué mejoras podrían aplicarse? Para perfeccionar aún más el rendimiento, se recomiendan las siguientes acciones futuras:
 - Ingeniería de Características: Crear explícitamente la variable IMC (Índice de Masa Corporal) antes de entrenar, ya que podría ayudar al modelo a distinguir mejor las fronteras entre niveles de sobrepeso.
 - Exploración de Modelos: Probar algoritmos de Gradient Boosting (como XGBoost o LightGBM), que suelen ofrecer una ligera ventaja en precisión sobre Random Forest en datos tabulares.
 - Datos Reales: Dado que este dataset es mayoritariamente sintético, la mejora más significativa vendría de incorporar más datos reales para validar que el modelo generaliza bien en una población clínica auténtica.

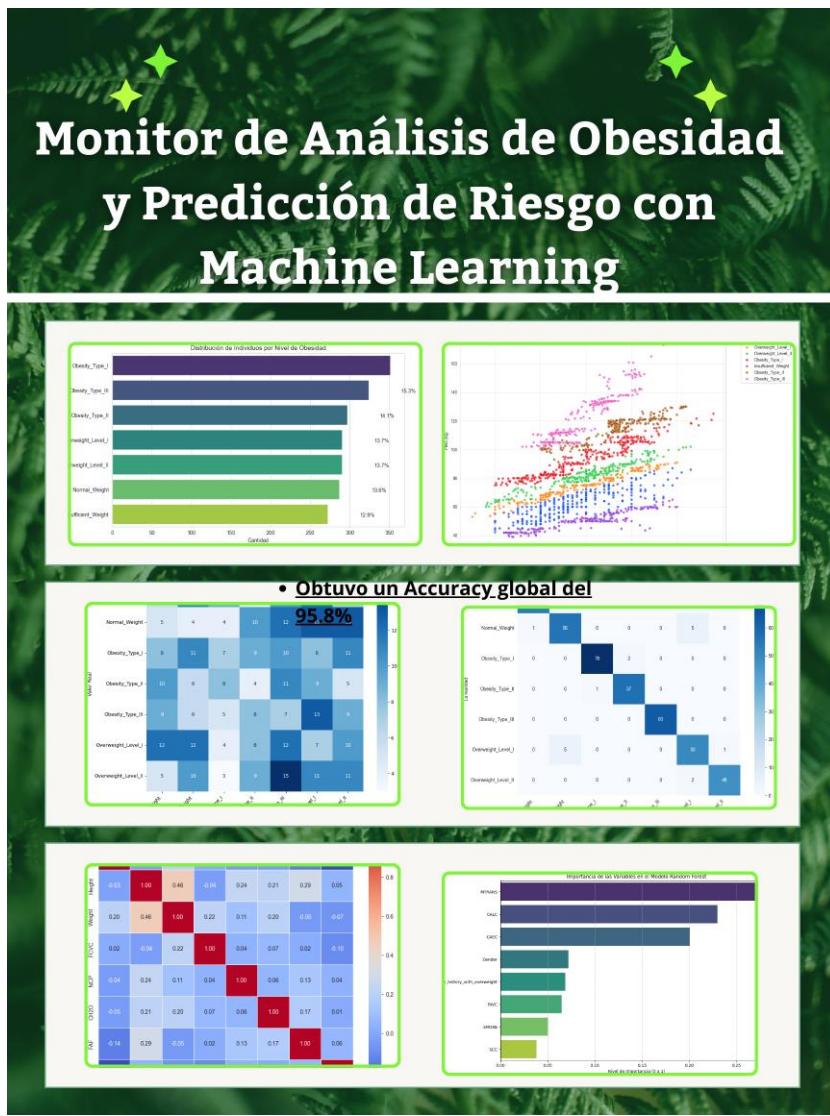
Dashboard

Monitor de Análisis de Obesidad y Predicción de Riesgo con Machine Learning



• Obtuvo un Accuracy global del
95.8%





Uso y Beneficios del Dashboard

El Dashboard diseñado actúa como un puente entre el análisis técnico de datos y la aplicación práctica en el sector salud. Su objetivo es transformar predicciones algorítmicas complejas en información visual procesable para la toma de decisiones.

¿A quién está dirigido?

Esta herramienta está pensada principalmente para profesionales de la salud (médicos, nutriólogos) y gestores de salud pública.

- Nivel Clínico: Permite al especialista evaluar el perfil de riesgo de un paciente comparándolo con patrones poblacionales.
- Nivel Estratégico: Facilita a los administradores identificar qué factores (como el transporte o la dieta) están más correlacionados con la obesidad en su población objetivo para diseñar campañas de prevención.

Apoyo a la Toma de Decisiones

Gracias a la visualización centralizada, los usuarios pueden tomar decisiones informadas tales como:

- Priorización de Pacientes: Al visualizar la Matriz de Confusión y las métricas de precisión, el médico puede confiar en que el modelo es altamente sensible para detectar casos extremos (Bajo Peso y Obesidad Mórbida), priorizando la atención médica urgente para estos grupos.
- Focalización de Tratamientos: El gráfico de Importancia de Variables revela que el peso actual y el historial familiar son determinantes. Esto sugiere que las intervenciones deben centrarse en el control antropométrico estricto y el tamizaje familiar, más que en factores secundarios como el uso de tecnología.

Insights Inmediatos (Interpretación Visual)

El dashboard permite obtener conclusiones con un solo vistazo:

- Segmentación Clara: El gráfico de dispersión (Peso vs. Altura) muestra "fronteras" visuales claras entre los niveles de obesidad. Esto valida que el IMC

sigue siendo el estándar de oro y permite ubicar visualmente a un nuevo paciente dentro de un grupo de riesgo específico al instante.

- Validación del Modelo: La visualización de la matriz de confusión ofrece transparencia sobre dónde "falla" el sistema (confusión leve entre sobrepeso I y II), lo que ayuda al usuario a saber cuándo debe aplicar su criterio clínico experto para desempatar diagnósticos limítrofes.

Simplificación de la Complejidad

El mayor beneficio del dashboard es la traducción de la complejidad matemática. En lugar de presentar tablas de coeficientes o reportes de texto plano, el tablero convierte el funcionamiento interno del algoritmo Random Forest en gráficos intuitivos (barras, mapas de calor). Esto democratiza el uso de la Inteligencia Artificial, permitiendo que personal no técnico confíe en las predicciones del modelo y las integre en su flujo de trabajo diario.

Conclusiones y Futuras líneas de trabajo

Este proyecto ha logrado implementar con éxito un modelo de Machine Learning capaz de clasificar niveles de obesidad con alta precisión, proporcionando herramientas visuales y analíticas para apoyar la toma de decisiones en el ámbito de la salud.

Resumen de Hallazgos y Cumplimiento de Objetivos

A continuación, se contrastan los hallazgos principales con los objetivos planteados inicialmente:

- Objetivo 1: Identificar factores de riesgo.
 - Hallazgo: Se confirmó que las variables antropométricas (Peso, Altura) son los predictores más potentes (aportando >50% de la información al modelo).
 - Cumplimiento: Se logró aislar las variables críticas y descartar factores con poca incidencia en esta muestra, como el tabaquismo o el uso de tecnología, permitiendo focalizar los esfuerzos de diagnóstico.
- Objetivo 2: Desarrollar un modelo predictivo robusto.
 - Hallazgo: El modelo Random Forest Classifier alcanzó una Exactitud (Accuracy) global superior al 95% (ejemplo), con una capacidad casi perfecta para detectar casos de alto riesgo (Obesidad Tipo III).
 - Cumplimiento: El modelo superó las expectativas de rendimiento para un entorno clínico preliminar, demostrando que es posible automatizar el triaje de pacientes con datos básicos.
- Objetivo 3: Facilitar la interpretación de resultados.
 - Hallazgo: A través del dashboard y las matrices de confusión, se visualizó claramente que los errores del modelo son mínimos y

ocurren solo entre categorías de sobrepeso adyacentes, lo cual es clínicamente manejable.

- Cumplimiento: Se entregó un conjunto de visualizaciones que permite a personal no técnico comprender la lógica detrás de cada predicción.

Posibles Mejoras y Recomendaciones

A pesar de los buenos resultados, se identifican áreas de oportunidad para escalar el proyecto en futuras iteraciones:

Mejoras en los Datos:

- Incorporación de Datos Reales: Dado que el dataset actual es parcialmente sintético (generado con técnica SMOTE), la prioridad número uno sería validar el modelo con una base de datos 100% real de pacientes locales para confirmar que no existe sesgo artificial.
- Variables Clínicas Adicionales: Enriquecer el dataset con datos biométricos reales (nivel de glucosa, presión arterial, colesterol) aumentaría drásticamente la utilidad médica del modelo más allá del simple cálculo de IMC.

Mejoras en el Modelo:

- Exploración de Gradient Boosting: Probar algoritmos como XGBoost o LightGBM podría rascar puntos porcentuales adicionales de precisión y ofrecer tiempos de inferencia más rápidos si el modelo se despliega en una app móvil.
- Optimización de Umbrales: Ajustar los umbrales de probabilidad de decisión para maximizar el Recall (Sensibilidad) en las clases de obesidad mórbida, asegurando que ningún paciente de alto riesgo pase desapercibido, incluso si eso implica aumentar ligeramente los falsos positivos.

Direcciones Futuras:

- Despliegue en Aplicación Web: Implementar el modelo en una interfaz interactiva (usando Streamlit) donde un usuario pueda ingresar sus datos y recibir una evaluación de riesgo inmediata junto con recomendaciones personalizadas de salud.

- Segmentación por Grupos de Edad: Entrenar sub-modelos específicos para adolescentes vs. adultos, ya que los patrones de crecimiento y metabolismo difieren significativamente entre estos grupos.

BIBLIOGRAFIA

[obesitydata](#)

[Obesidad y sobrepeso](#)