Prueba Intertrimestral

Nombre:
Apellidos:

Tiempo de la prueba: 2 horas
Asignatura: Desarrollo de Aplicaciones para la Visualización de Datos
Fecha: 16 de octubre de 2025

Instrucciones

- Escribe **código limpio y autoexplicativo**.
- Se pueden utilizar **los materiales de clase**.
- Se puede utilizar **internet** para búsqueda de dudas y documentación.
- **No se puede utilizar ningún tipo de LLM** (ChatGPT, Copilot, Gemini, etc.).
- **No se puede utilizar mensajería instantánea**.
- Al finalizar, **sube tu notebook a GitHub** y envía el enlace del fichero en el siguiente formulario:

https://forms.office.com/e/LFVwu9z6uQ

Dataset “Life Style Data”

El dataset contiene información sobre hábitos de vida (alimentación, sueño, ejercicio, consumo de tabaco o alcohol, etc.) y medidas de salud (IMC, presión arterial, etc.) para diferentes individuos.

Puedes descargarlo del repositorio de la asignatura y un .txt con la descripción de las variables.

Inicialización de librerías

In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score, root_mean_squared_err
from itertools import combinations
```

Ejercicio 1 — Programación (2 puntos)

a) (0.6 pts) Crea una función imc(weight, height) que devuelva el índice de masa corporal (IMC), redondeado a dos decimales.

In [12]:
```
def imc(weight, height):
    #Asumiendo que nos dan el valor en metros y el peso en kilos
    valor = weight /(height*height)
    return round(valor,2)

imc(70,1.75)
```

Out[12]: 22.86

b) (0.6 pts) Crea una función saludable(imcValue) que devuelva "Saludable" si el IMC está entre 18.5 y 24.9, "Bajo peso" si es menor de 18.5, o "Sobrepeso" si es mayor de 24.9.

In [99]:
```
def saludable(imcValue):
    if imcValue < 18.5:
        variable = "Bajo peso"
        return variable
    if imcValue>= 18.5 and imcValue<=25.9:
        variable = "Saludable"
        return variable
    else:
        variable = "Sobrepeso"
        return variable

saludable(imc(70,1.75))
```

Out[99]: 'Saludable'

c) (0.6 pts) Genera un DataFrame llamado imcData que contenga al menos 10 registros con las siguientes columnas:

- peso (en kilogramos)
- altura (en metros)
- imc (calculado usando la función imc(weight, height) creada anteriormente)

Puedes generar los datos de peso y altura manualmente, mediante listas, o de forma aleatoria utilizando numpy (np.random.uniform o similar).

Añade una cuarta columna llamada categoria, cuyos valores provengan de la función saludable(imcValue).

Muestra las primeras filas del DataFrame resultante y comprueba que los tipos de datos son correctos.

In [105…
```
# Parámetros iniciales
pesos = [70,80,90,75,85,78,62,58,92,101]
alturas = [1.69,1.80,1.67,1.87,1.74,1.67,2.03,1.55,1.76,1.67]

# Lista para almacenar los resultados
resultados = []
categoria =[]
```

```
for n, peso in enumerate(pesos):
    resultados.append([peso,alturas[n],imc(peso,alturas[n]),saludable(imc(peso,a
# Crear el DataFrame
df = pd.DataFrame(resultados, columns=['peso', 'altura', 'imc','categoria'])

# Mostrar los 10 primeros resultados
print(df.head(10))
```

```
   peso  altura    imc  categoria
0    70    1.69  24.51  Saludable
1    80    1.80  24.69  Saludable
2    90    1.67  32.27  Sobrepeso
3    75    1.87  21.45  Saludable
4    85    1.74  28.08  Sobrepeso
5    78    1.67  27.97  Sobrepeso
6    62    2.03  15.05  Bajo peso
7    58    1.55  24.14  Saludable
8    92    1.76  29.70  Sobrepeso
9   101    1.67  36.21  Sobrepeso
```

d) (0.2 pts) ¿Cómo podrías integrar ambas funciones dentro de una clase llamada HealthTools que calcule el IMC y clasifique automáticamente a partir de listas de pesos y alturas?

In [109…

```
class HealthTools:
    def __init__(self, peso, altura):
        self.peso = peso
        self.altura = altura

    def imc(self,peso,altura):
        return round(peso /(altura*altura),2)

    def saludable(self, peso, altura):
        if self.imc(peso,altura) < 18.5:
            variable = "Bajo peso"
            return variable
        if imcValue>= 18.5 and imcValue<=25.9:
            self.imc(peso,altura) = "Saludable"
            return variable
        else:
            self.imc(peso,altura) = "Sobrepeso"
            return variable
```

```
  Cell In[109], line 14
    self.imc(peso,altura) = "Saludable"
    ^
SyntaxError: cannot assign to function call here. Maybe you meant '==' instead of
'='?
```

Ejercicio 2 — Exploración y visualización (3 puntos)

a) (0.75 pts) Carga el dataset desde el fichero CSV y guárdalo en un DataFrame llamado lifeStyleData. Muestra las 5 primeras filas, el número de filas y columnas.

In [229…

```
import pandas as pd
from pathlib import Path
```

```
def leer_fichero(ruta):
    p = Path(ruta)
    if not p.exists():
        raise FileNotFoundError(f"No existe el fichero: {p}")
    #hay que ver ahora que tipo de archivo es
    ext = p.suffix.lower()
    if ext == ".csv":
        return pd.read_csv(p)
    elif ext == ".json":
        return pd.read_json(p)
    elif ext in (".xlsx", ".xls"):
        return pd.read_excel(p)
    else:
        raise ValueError("Formato no valido, hay que usar .csv, .json o .xlsx/.x
data = leer_fichero("final_data.csv")
df = pd.DataFrame(data)
print(df.head(5))
```

```
     Age  Gender  Weight (kg)  Height (m)  Max_BPM  Avg_BPM  Resting_BPM  \
0  34.91    Male        65.27        1.62   188.58   157.65        69.05
1  23.37  Female        56.41        1.55   179.43   131.75        73.18
2  33.20  Female        58.98        1.67   175.04   123.95        54.96
3  38.69  Female        93.78        1.70   191.21   155.10        50.07
4  45.09    Male        52.42        1.88   193.58   152.88        70.84

   Session_Duration (hours)  Calories_Burned Workout_Type  ...  \
0                      1.00          1080.90     Strength  ...
1                      1.37          1809.91         HIIT  ...
2                      0.91           802.26       Cardio  ...
3                      1.10          1450.79         HIIT  ...
4                      1.08          1166.40     Strength  ...

   cal_from_macros  pct_carbs  protein_per_kg   pct_HRR  pct_maxHR  \
0          2139.59   0.500432        1.624789  0.741237   0.835985
1          1711.65   0.500850        1.514093  0.551247   0.734270
2          1965.92   0.500610        1.663445  0.574534   0.708124
3          1627.28   0.499533        0.862017  0.744155   0.811150
4          2659.23   0.500581        2.538153  0.668405   0.789751

   cal_balance  lean_mass_kg  expected_burn  Burns Calories (per 30 min)_bc  \
0       725.10     47.777394       685.1600                    7.260425e+19
1      -232.91     40.809803       978.6184                    1.020506e+20
2       805.74     44.635580       654.5266                    1.079607e+20
3      1206.21     63.007432       773.6300                    8.987921e+19
4       303.60     43.347504       711.4176                    5.264685e+19

   Burns_Calories_Bin
0              Medium
1                High
2                High
3                High
4                 Low

[5 rows x 54 columns]
```

b) (0.75 pts) Crea una función describeData(dataFrame) que devuelva para cada columna: tipo de dato, número de valores nulos y porcentaje de nulos.

In [231…

```python
def describeData(dataFrame):
    descripcion = {}
    for columna in dataFrame.columns:
        tipo_dato = dataFrame[columna].dtype
        nulos = dataFrame[columna].isnull().sum()
        porcentaje_nulos = (nulos / len(dataFrame)) * 100
        descripcion[columna] = {
            'tipo': tipo_dato,
            'nulos': nulos,
            'porcentaje_nulos': porcentaje_nulos
        }
    return descripcion

describeData(df)
```

Out[231…


```
{'Age': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Gender': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Weight (kg)': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Height (m)': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Max_BPM': {'tipo': dtype('float64'), 'nulos': 1, 'porcentaje_nulos': 0.005},
 'Avg_BPM': {'tipo': dtype('float64'), 'nulos': 1, 'porcentaje_nulos': 0.005},
 'Resting_BPM': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Session_Duration (hours)': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Calories_Burned': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Workout_Type': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Fat_Percentage': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Water_Intake (liters)': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Workout_Frequency (days/week)': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Experience_Level': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'BMI': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Daily meals frequency': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Physical exercise': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Carbs': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Proteins': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Fats': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Calories': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'meal_name': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'meal_type': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'diet_type': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'sugar_g': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'sodium_mg': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'cholesterol_mg': {'tipo': dtype('float64'),
  'nulos': 1,
  'porcentaje_nulos': 0.005},
 'serving_size_g': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'cooking_method': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'prep_time_min': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'cook_time_min': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'rating': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
```

```
 'Name of Exercise': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Sets': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Reps': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Benefit': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Burns Calories (per 30 min)': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Target Muscle Group': {'tipo': dtype('O'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Equipment Needed': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Difficulty Level': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Body Part': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Type of Muscle': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'Workout': {'tipo': dtype('O'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'BMI_calc': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'cal_from_macros': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'pct_carbs': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'protein_per_kg': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'pct_HRR': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'pct_maxHR': {'tipo': dtype('float64'), 'nulos': 0, 'porcentaje_nulos': 0.0},
 'cal_balance': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'lean_mass_kg': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'expected_burn': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Burns Calories (per 30 min)_bc': {'tipo': dtype('float64'),
  'nulos': 0,
  'porcentaje_nulos': 0.0},
 'Burns_Calories_Bin': {'tipo': dtype('O'),
  'nulos': 0,
  'porcentaje_nulos': 0.0}}
```


c) (0.75 pts) Usa describeData(lifeStyleData) y comenta brevemente (2–3 líneas) qué variables parecen tener más valores faltantes.

Las variables con más nulos son: MaxBPM, AvgBP, cholesterol_mg con 1 nulo cada uno

d) (0.75 pts) Realiza un gráfico que muestre las correlaciones entre variables numéricas. ¿Hay alguna correlación fuerte o inesperada?

In [327…

```python
import plotly.express as px

print(df.corr())
px.imshow(df.corr())
##he ido eliminando 1 por 1 por eso no salen los pd.drop()
##se que lo suyo habría sido convertir las variables categoricas en numericas /(
##mucho tiempo y las he dropeado simplemente
```

	Age	Weight (kg)	Height (m)	Max_BPM
Age	1.000000	-0.044077	-0.037096	-0.028788
Weight (kg)	-0.044077	1.000000	0.353763	0.066906
Height (m)	-0.037096	0.353763	1.000000	-0.014376
Max_BPM	-0.028788	0.066906	-0.014376	1.000000
Avg_BPM	0.039278	0.016731	-0.012187	-0.027995
Resting_BPM	0.013387	-0.033578	0.009897	0.035500
Session_Duration (hours)	-0.025410	-0.002275	0.006783	0.009761
Calories_Burned	-0.021396	-0.001994	0.009211	0.003114
Fat_Percentage	-0.025056	0.778875	-0.156586	0.071910
Water_Intake (liters)	0.037569	0.397971	0.398041	0.041655
Workout_Frequency (days/week)	0.007182	-0.003894	0.007241	-0.038316
Experience_Level	-0.022485	0.016531	0.015537	-0.003648
BMI	-0.016940	0.855599	-0.167011	0.073527
Daily meals frequency	0.037914	0.041891	-0.013809	-0.024548
Physical exercise	0.021586	-0.008223	0.014574	-0.013665
Carbs	0.051966	-0.003567	0.008214	-0.010580
Proteins	0.052172	-0.003493	0.007845	-0.010511
Fats	0.052071	-0.003620	0.007996	-0.010462
Calories	-0.041096	0.978273	0.346939	0.061164
sugar_g	0.011649	-0.007367	-0.019130	0.045741
sodium_mg	0.044882	-0.013639	0.009843	0.020658
cholesterol_mg	0.012708	0.019546	0.028076	-0.016904
serving_size_g	0.008652	0.037870	0.037035	-0.035807
prep_time_min	-0.011140	-0.021312	0.045705	0.004511
cook_time_min	-0.009893	-0.038063	-0.018457	-0.016428
rating	-0.010725	-0.015728	0.016786	-0.056228
Sets	0.020391	0.008588	-0.017659	-0.030311
Reps	-0.028396	-0.026516	0.003823	-0.041839
Burns Calories (per 30 min)	0.005641	0.090584	0.065279	-0.022803
BMI_calc	-0.016938	0.855598	-0.167014	0.073523
cal_from_macros	0.052045	-0.003569	0.008076	-0.010532
pct_carbs	-0.005022	-0.001333	0.012513	-0.004345
protein_per_kg	0.067281	-0.726714	-0.246187	-0.055944
pct_HRR	0.044258	-0.016656	-0.012385	-0.511863
pct_maxHR	0.048140	-0.023117	-0.005411	-0.559436
cal_balance	-0.015984	0.736698	0.254356	0.043805
lean_mass_kg	-0.050547	0.981897	0.477397	0.060160
expected_burn	-0.025910	0.024249	0.029134	0.003508
Burns Calories (per 30 min)_bc	-0.020948	0.253594	0.103753	-0.029654

	Avg_BPM	Resting_BPM
Age	0.039278	0.013387
Weight (kg)	0.016731	-0.033578
Height (m)	-0.012187	0.009897
Max_BPM	-0.027995	0.035500
Avg_BPM	1.000000	0.063000
Resting_BPM	0.063000	1.000000
Session_Duration (hours)	0.018598	-0.017554
Calories_Burned	0.008069	-0.001633
Fat_Percentage	0.009026	-0.042021
Water_Intake (liters)	0.006197	0.011676
Workout_Frequency (days/week)	-0.020784	-0.005480
Experience_Level	-0.005485	-0.000831
BMI	0.028926	-0.040734
Daily meals frequency	0.036676	0.040364
Physical exercise	-0.025418	0.015181
Carbs	0.033193	0.042304
Proteins	0.033485	0.042601
Fats	0.033132	0.042740

```
Calories                          0.014462    -0.034669
sugar_g                          -0.046026     0.034085
sodium_mg                         0.017820     0.051654
cholesterol_mg                   -0.013425    -0.030551
serving_size_g                    0.017661    -0.004578
prep_time_min                     0.004518    -0.008292
cook_time_min                    -0.022993    -0.033161
rating                           -0.018772    -0.000528
Sets                             -0.046223     0.002984
Reps                             -0.031362     0.001295
Burns Calories (per 30 min)      -0.073358     0.037717
BMI_calc                          0.028924    -0.040731
cal_from_macros                   0.033237     0.042500
pct_carbs                        -0.005006    -0.016604
protein_per_kg                    0.018188     0.065700
pct_HRR                           0.856728    -0.088468
pct_maxHR                         0.841881     0.034978
cal_balance                       0.005250    -0.024921
lean_mass_kg                      0.012048    -0.030811
expected_burn                    -0.005337    -0.004718
Burns Calories (per 30 min)_bc   -0.058322    -0.000931

                                 Session_Duration (hours)  Calories_Burned  \
Age                                             -0.025410        -0.021396
Weight (kg)                                     -0.002275        -0.001994
Height (m)                                       0.006783         0.009211
Max_BPM                                          0.009761         0.003114
Avg_BPM                                          0.018598         0.008069
Resting_BPM                                     -0.017554        -0.001633
Session_Duration (hours)                         1.000000         0.814368
Calories_Burned                                  0.814368         1.000000
Fat_Percentage                                  -0.034861        -0.033812
Water_Intake (liters)                            0.287751         0.262731
Workout_Frequency (days/week)                    0.637626         0.582787
Experience_Level                                 0.758465         0.697116
BMI                                             -0.004283        -0.004494
Daily meals frequency                            0.022529         0.028158
Physical exercise                                0.048867         0.049113
Carbs                                            0.010626         0.011929
Proteins                                         0.010398         0.011602
Fats                                             0.010476         0.011769
Calories                                         0.051474         0.047981
sugar_g                                         -0.010797        -0.003044
sodium_mg                                       -0.014477        -0.009472
cholesterol_mg                                   0.092563         0.072279
serving_size_g                                   0.045158         0.029458
prep_time_min                                   -0.003194        -0.002823
cook_time_min                                   -0.055469        -0.061617
rating                                           0.022199         0.025625
Sets                                             0.011456         0.024164
Reps                                             0.033300         0.043237
Burns Calories (per 30 min)                     -0.012035         0.006562
BMI_calc                                        -0.004282        -0.004495
cal_from_macros                                  0.010536         0.011817
pct_carbs                                        0.007848         0.009419
protein_per_kg                                  -0.042841        -0.035982
pct_HRR                                          0.017835         0.010506
pct_maxHR                                        0.010953         0.006106
cal_balance                                     -0.528619        -0.660560
lean_mass_kg                                     0.025165         0.023134
```

```
expected_burn                                0.944115         0.773932
Burns Calories (per 30 min)_bc              -0.030851        -0.013504

                            Fat_Percentage  Water_Intake (liters)  ...  \
Age                              -0.025056               0.037569  ...
Weight (kg)                       0.778875               0.397971  ...
Height (m)                       -0.156586               0.398041  ...
Max_BPM                           0.071910               0.041655  ...
Avg_BPM                           0.009026               0.006197  ...
Resting_BPM                      -0.042021               0.011676  ...
Session_Duration (hours)         -0.034861               0.287751  ...
Calories_Burned                  -0.033812               0.262731  ...
Fat_Percentage                    1.000000               0.185207  ...
Water_Intake (liters)             0.185207               1.000000  ...
Workout_Frequency (days/week)    -0.039340               0.241269  ...
Experience_Level                 -0.033251               0.312308  ...
BMI                               0.902341               0.214091  ...
Daily meals frequency             0.052791               0.034503  ...
Physical exercise                -0.009744               0.091540  ...
Carbs                            -0.012570               0.045370  ...
Proteins                         -0.012415               0.045209  ...
Fats                             -0.012606               0.045198  ...
Calories                          0.759601               0.411713  ...
sugar_g                          -0.003715               0.051388  ...
sodium_mg                        -0.017084               0.012098  ...
cholesterol_mg                    0.004857               0.023714  ...
serving_size_g                    0.017901               0.005520  ...
prep_time_min                    -0.042992               0.032287  ...
cook_time_min                    -0.016259              -0.008350  ...
rating                           -0.026013               0.037446  ...
Sets                              0.006836               0.005125  ...
Reps                             -0.033799              -0.032642  ...
Burns Calories (per 30 min)       0.054499               0.026161  ...
BMI_calc                          0.902341               0.214090  ...
cal_from_macros                  -0.012551               0.045292  ...
pct_carbs                        -0.003533               0.006100  ...
protein_per_kg                   -0.603368              -0.281664  ...
pct_HRR                          -0.020686              -0.019804  ...
pct_maxHR                        -0.030786              -0.018205  ...
cal_balance                       0.594499               0.126435  ...
lean_mass_kg                      0.659078               0.435249  ...
expected_burn                    -0.018944               0.281376  ...
Burns Calories (per 30 min)_bc    0.178974               0.072807  ...

                             BMI_calc  cal_from_macros  pct_carbs  \
Age                         -0.016938         0.052045  -0.005022
Weight (kg)                  0.855598        -0.003569  -0.001333
Height (m)                  -0.167014         0.008076   0.012513
Max_BPM                      0.073523        -0.010532  -0.004345
Avg_BPM                      0.028924         0.033237  -0.005006
Resting_BPM                 -0.040731         0.042500  -0.016604
Session_Duration (hours)    -0.004282         0.010536   0.007848
Calories_Burned             -0.004495         0.011817   0.009419
Fat_Percentage               0.902341        -0.012551  -0.003533
Water_Intake (liters)        0.214090         0.045292   0.006100
Workout_Frequency (days/week)  0.000506       0.024608  -0.004391
Experience_Level             0.016362         0.031710   0.003436
BMI                          1.000000        -0.003704  -0.006589
Daily meals frequency        0.058117         0.127032   0.006843
Physical exercise           -0.010619         0.644115   0.002320
```

```
Carbs                          -0.003755         0.999924   0.011611
Proteins                       -0.003523         0.999803  -0.010864
Fats                           -0.003716         0.999848  -0.013638
Calories                        0.837554         0.020660   0.000037
sugar_g                         0.000919         0.065572  -0.009679
sodium_mg                      -0.020860         0.014638  -0.002842
cholesterol_mg                  0.000350        -0.046142   0.002136
serving_size_g                  0.026001         0.019376  -0.002215
prep_time_min                  -0.039054         0.008572   0.004458
cook_time_min                  -0.028071         0.005645   0.004070
rating                         -0.030531         0.026686   0.000855
Sets                            0.022481         0.009577   0.012227
Reps                           -0.029888        -0.074847  -0.003418
Burns Calories (per 30 min)     0.062792        -0.002718  -0.000759
BMI_calc                        1.000000        -0.003697  -0.006591
cal_from_macros                -0.003697         1.000000  -0.000460
pct_carbs                      -0.006591        -0.000460   1.000000
protein_per_kg                 -0.634557         0.623063  -0.010057
pct_HRR                        -0.004671         0.023109   0.000655
pct_maxHR                      -0.015102         0.031472  -0.001498
cal_balance                     0.632669         0.007297  -0.006534
lean_mass_kg                    0.769326        -0.001817   0.000514
expected_burn                   0.012711         0.010146   0.008081
Burns Calories (per 30 min)_bc  0.211483         0.044494  -0.001778

                               protein_per_kg   pct_HRR  pct_maxHR  \
Age                                 0.067281  0.044258   0.048140
Weight (kg)                        -0.726714 -0.016656  -0.023117
Height (m)                         -0.246187 -0.012385  -0.005411
Max_BPM                            -0.055944 -0.511863  -0.559436
Avg_BPM                             0.018188  0.856728   0.841881
Resting_BPM                         0.065700 -0.088468   0.034978
Session_Duration (hours)           -0.042841  0.017835   0.010953
Calories_Burned                    -0.035982  0.010506   0.006106
Fat_Percentage                     -0.603368 -0.020686  -0.030786
Water_Intake (liters)              -0.281664 -0.019804  -0.018205
Workout_Frequency (days/week)      -0.035525  0.002299   0.002352
Experience_Level                   -0.050977  0.004123  -0.000211
BMI                                -0.634562 -0.004671  -0.015103
Daily meals frequency               0.042696  0.043344   0.045865
Physical exercise                   0.388480 -0.022156  -0.014987
Carbs                               0.622890  0.023128   0.031464
Proteins                            0.623177  0.023296   0.031667
Fats                                0.623019  0.022945   0.031341
Calories                           -0.701397 -0.015755  -0.022025
sugar_g                             0.052692 -0.069611  -0.066342
sodium_mg                           0.018502 -0.001775   0.002894
cholesterol_mg                     -0.043246  0.000532  -0.002662
serving_size_g                     -0.020717  0.033493   0.033093
prep_time_min                       0.025711  0.002905  -0.000354
cook_time_min                       0.047463 -0.008021  -0.009544
rating                              0.026243  0.010681   0.016007
Sets                               -0.002462 -0.020299  -0.020185
Reps                               -0.030575 -0.003810  -0.004295
Burns Calories (per 30 min)        -0.080888 -0.055057  -0.048039
BMI_calc                           -0.634557 -0.004671  -0.015102
cal_from_macros                     0.623063  0.023109   0.031472
pct_carbs                          -0.010057  0.000655  -0.001498
protein_per_kg                      1.000000  0.033100   0.045408
pct_HRR                             0.033100  1.000000   0.988485
```

```
pct_maxHR                          0.045408  0.988485   1.000000
cal_balance                       -0.502131 -0.019161  -0.020808
lean_mass_kg                      -0.722151 -0.018484  -0.023716
expected_burn                     -0.063268 -0.000937  -0.005444
Burns Calories (per 30 min)_bc    -0.157793 -0.036191  -0.033204

                                cal_balance  lean_mass_kg  expected_burn  \
Age                               -0.015984     -0.050547      -0.025910
Weight (kg)                        0.736698      0.981897       0.024249
Height (m)                         0.254356      0.477397       0.029134
Max_BPM                            0.043805      0.060160       0.003508
Avg_BPM                            0.005250      0.012048      -0.005337
Resting_BPM                       -0.024921     -0.030811      -0.004718
Session_Duration (hours)          -0.528619      0.025165       0.944115
Calories_Burned                   -0.660560      0.023134       0.773932
Fat_Percentage                     0.594499      0.659078      -0.018944
Water_Intake (liters)              0.126435      0.435249       0.281376
Workout_Frequency (days/week)     -0.344260      0.024742       0.618920
Experience_Level                  -0.419532      0.048707       0.735709
BMI                                0.632670      0.769327       0.012711
Daily meals frequency              0.015081      0.034477       0.035597
Physical exercise                 -0.008609     -0.001513       0.061516
Carbs                              0.007227     -0.001798       0.010244
Proteins                           0.007496     -0.001785       0.009998
Fats                               0.007277     -0.001870       0.010077
Calories                           0.718214      0.962934       0.076448
sugar_g                            0.002135     -0.009695      -0.037596
sodium_mg                         -0.006686     -0.011786      -0.024137
cholesterol_mg                    -0.034744      0.023268       0.075105
serving_size_g                     0.010532      0.041091       0.040220
prep_time_min                     -0.013403     -0.015123      -0.021506
cook_time_min                      0.010019     -0.043574      -0.057772
rating                            -0.028924     -0.009525       0.017130
Sets                              -0.007059      0.009546       0.159466
Reps                              -0.046588     -0.023183       0.144142
Burns Calories (per 30 min)        0.065790      0.097562       0.311900
BMI_calc                           0.632669      0.769326       0.012711
cal_from_macros                    0.007297     -0.001817       0.010146
pct_carbs                         -0.006534      0.000514       0.008081
protein_per_kg                    -0.502131     -0.722151      -0.063268
pct_HRR                           -0.019161     -0.018484      -0.000937
pct_maxHR                         -0.020808     -0.023716      -0.005444
cal_balance                        1.000000      0.707662      -0.481679
lean_mass_kg                       0.707662      1.000000       0.052819
expected_burn                     -0.481679      0.052819       1.000000
Burns Calories (per 30 min)_bc     0.197965      0.253178       0.231644

                                Burns Calories (per 30 min)_bc
Age                                                  -0.020948
Weight (kg)                                           0.253594
Height (m)                                            0.103753
Max_BPM                                              -0.029654
Avg_BPM                                              -0.058322
Resting_BPM                                          -0.000931
Session_Duration (hours)                             -0.030851
Calories_Burned                                      -0.013504
Fat_Percentage                                        0.178974
Water_Intake (liters)                                 0.072807
Workout_Frequency (days/week)                         0.011216
Experience_Level                                      0.019175
```

BMI	0.211487
Daily meals frequency	0.044156
Physical exercise	0.065746
Carbs	0.044494
Proteins	0.044491
Fats	0.044479
Calories	0.250862
sugar_g	-0.070094
sodium_mg	0.000813
cholesterol_mg	-0.041557
serving_size_g	0.000441
prep_time_min	-0.059320
cook_time_min	-0.007256
rating	-0.017266
Sets	0.314912
Reps	0.268278
Burns Calories (per 30 min)	0.814548
BMI_calc	0.211483
cal_from_macros	0.044494
pct_carbs	-0.001778
protein_per_kg	-0.157793
pct_HRR	-0.036191
pct_maxHR	-0.033204
cal_balance	0.197965
lean_mass_kg	0.253178
expected_burn	0.231644
Burns Calories (per 30 min)_bc	1.000000

[39 rows x 39 columns]

Ejercicio 3 — Regresión básica (2 puntos)

Queremos predecir una variable de salud continua, por ejemplo bmi.

a) (0.25 pts) Define targetVar = "bmi" y selecciona 2–3 variables predictoras (por ejemplo sleepHours, activityLevel, calories). Guarda las columnas predictoras en X y la variable objetivo en y.

In [332…
```
## aqui voy a iniciar mis propias variables porque creo que el anterior no lo he
## estar mal desde un punto de vista numerico pero están bien desde un punto de
data = {'bmi': [25.5, 22.1, 28.9, 31.0, 24.5],
        'sleepHours': [7, 8, 6, 5, 7],
        'activityLevel': [60, 45, 30, 20, 50],
        'calories': [2000, 1800, 2500, 3000, 2200]}
df = pd.DataFrame(data)

targetVar = "bmi"

predictor_vars = ['sleepHours', 'activityLevel', 'calories']

X = df[predictor_vars]
Y = df[targetVar]
```

b) (0.25 pts) Divide los datos en entrenamiento (70 %) y prueba (30 %).

In [335…
```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.3)
```

c) (0.75 pts) Entrena un modelo de regresión lineal (LinearRegression) y calcula el R^2 y el MSE.

In [348…
```
from sklearn.metrics import r2_score, mean_squared_error

model = LinearRegression()
model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)

r2 = r2_score(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)

print("R² del modelo:", r2)
print("MSE del modelo:", mse)
```

```
R² del modelo: -23.142133799398824
MSE del modelo: 279.08306672105016
```

d) (0.75 pts) Entrena un RandomForestRegressor con los mismos datos y compara resultados. ¿Cuál tiene mejor rendimiento?

In [359…
```
from sklearn.ensemble import RandomForestRegressor
model2 = RandomForestRegressor()
model2.fit(X_train, Y_train)
```

```
Y_pred = model2.predict(X_test)

r2 = r2_score(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)

print("R² del modelo:", r2)
print("MSE del modelo:", mse)
```

```
R² del modelo: 0.4631444636678197
MSE del modelo: 6.206049999999999
```

funciona mejor en el caso de este modelo R^2 y una MSE mucho menor. (obviamente los datos son inventados)

Ejercicio 4 — Clasificación sencilla (2 puntos)

Crea una nueva variable binaria que indique si el IMC está dentro del rango saludable.

a) (0.25 pts) Crea una nueva columna bmiHealthy donde el valor sea 1 si bmi está entre 18.5 y 24.9, y 0 en caso contrario.

In [ ]:

b) (0.75 pts) Entrena un modelo de regresión logística (LogisticRegression) para predecir bmiHealthy usando algunas variables de estilo de vida (por ejemplo sleepHours, activityLevel, calories).

In [ ]:

c) (0.25 pts) Calcula la exactitud (accuracy) del modelo y muestra la matriz de confusión.

In [ ]:

d) (0.75 pts) Explica qué variable parece influir más según el modelo.

In [ ]:

Ejercicio 5 — Conclusión (1 punto)

En un máximo de 300 palabras, resume:

- Qué hábitos parecen tener mayor relación con el estado de salud.

  Aqui simplemente habría que ver los valores con una mayor correlación y luego un mayor coeficiente en una regresion lineal.

- Qué modelo fue más eficaz y por qué.
- Qué podrías mejorar si tuvieras más tiempo o datos.

primero haria una limpia como dios manda haciendo replace de los valores nulos, y sobretodo crearia dummies para las variables categoricas, también haría cross validation

con los datos. Intentaría buscar variables que estuvieran demasiado corrleacionadas y las eliminaría o crearía variables nuevas combinanda ambas dos.