1. Descripción del dataset

El dataset a tratar en la práctica es https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset.

Se trata de un dataset con 14 variables, las cuales estan asociadas a pancientes con síntomas coronarios y cuyo objetivo es la predicción de la probabilidad (mayor o menor) de tener un ataque al corazón bajo esos síntomas. El dataset es de mucha relevancia, dado que poder entrerar un modelo que anticipe esta circunstancia podría ayudar a salvar muchas vidas.

El dataset consta de las siguientes variables:

- **age** : Edad del paciente
- **sex**: Sexo del paciente codificado como 0 o 1. Se desconoce su traducción a Hombre o Mujer.
- **cp** : Chest Pain type chest pain type. Tipo de dolor en el pecho. Puede tomar los valores:
  - Valor 0: typical angina
  - Valor 1: atypical angina
  - Valor 2: non-anginal pain
  - Valor 3: asymptomatic

    > Aunque en el dataset de referenca de Kaggle de indica que el rango de valores de la variable es [1..4], analizando el fichero se observa que es [0..3]
- **trtbps** : resting blood pressure (in mm Hg). Presion sanguinea en reposo.
- **chol** : cholestoral in mg/dl fetched via BMI sensor. Colesterol en sangre.
- **fbs** : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false). Azugar en sangre en ayunas por encima de 129 mg/dl. Codificado como 1 Verdadero, 0 Falso.
- **restecg** : resting electrocardiographic results. Resultados del electrocardiograma en reposo. Puede tomar los siguientes valores:
  - Valor 0: normal
  - Vaslor 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
  - Valor 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.
- **thalachh** : maximum heart rate achieved. Frecuencia cardiaca máxima.

- **exng**: "exercise induced angina". Angina inducida por el ejercicio. Codificado como 1 "Si" 0 "No".
- **caa**: number of major vessels. Numero de vasos sanguineos mayores. Codificado de 0 a 4.

> En la descripción del dataset de Kaggle se indica que que el rango de la variable es 0..3, sin embargo observando el fichero, se codifica de 0..4. Esto es mas coherente, ya que los vasos mayores del corazón son 5.

- **target** : Variable objetivo. 0 = menor posibilidad de ataque al corazón 1 = mayor posibilidad de ataque al corazon.

El dataset consta de otras tres variables que no están descritas y no conocemos su significado (oldpoeak, slp y thall) que no utilizaremos por pruedencia.

A continuación, vamos a visualizar los primeros datos del dataset

In [190…
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [191…
```
df = pd.read_csv("./datos/heart.csv")
```

In [192…
```
df.head()
```

Out[192]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Borramos las columnas que no vamos a utlizar

In [193…
```
df = df.drop(["oldpeak", "slp", "thall"], axis=1)
```

2. Integración y selección

Vamos a integrar los ficheros que contienen las descriciones de los campos categoricos del dataset, de manera que sea más fácil su interpretación y trabajo con los datos.

Después de la integración de cada fichero, borraremos la columna de cruce del fichero integrado, para no duplicar la columna.

Exang

In [194…
```
df_exang = pd.read_csv("./datos/exang.csv")
```

In [195…
```
df = df.merge(df_exang, how="left", left_on="exng", right_on="id_exang").drop(["
```

Chest pain

In [196…
```
df_cp = pd.read_csv("./datos/chest_pain.csv")
```

In [197…
```
df = df.merge(df_cp, left_on="cp", right_on="id_cp", how="left").drop(["id_cp"],
```

Visualizamos el dataset final

In [198…
```
df
```

Out[198]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	caa	output	desc_exang	
0	63	1	3	145	233	1	0	150	0	0	1	no	asym
1	37	1	2	130	250	0	1	187	0	0	1	no	nor
2	41	0	1	130	204	0	0	172	0	0	1	no	
3	56	1	1	120	236	0	1	178	0	0	1	no	
4	57	0	0	120	354	0	1	163	1	0	1	yes	typic
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0	0	yes	typic
299	45	1	3	110	264	0	1	132	0	0	0	no	asym
300	68	1	0	144	193	1	1	141	0	2	0	no	typic
301	57	1	0	130	131	0	1	115	1	1	0	yes	typic
302	57	0	1	130	236	0	0	174	0	1	0	no	

303 rows × 13 columns

3. Limpieza de los datos.

NULOS

Obenemos un listado de las columnas que tienen nulos y el porcentaje que representan sobre el total de datos.

In [199…
```
nulos =  [(column, round(df[column].isnull().sum() / len(df[column])*100))
          for column in df.columns
          if  df[column].isnull().sum()!=0]
nulos
```

Out[199]: []

No hay datos nulos

PERDIDOS

Vamos a identificar valores extraños que puedan significar perdida de datos:

age

In [203…
```
df.age.describe()
```

Out[203]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

Minimo y máximo en rangos coherentes. La información parece correcta.

sex

In [202…
```
df.sex.unique()
```

Out[202]:
```
array([1, 0], dtype=int64)
```

Correcta

cp

In [205…
```
df.cp.unique()
```

Out[205]:
```
array([3, 2, 1, 0], dtype=int64)
```

Valores en rango. Parece correcta.

trtbps

In [206…
```
df.trtbps.describe()
```

Out[206]:
```
count    303.000000
mean     131.623762
std       17.538143
min       94.000000
25%      120.000000
50%      130.000000
75%      140.000000
max      200.000000
Name: trtbps, dtype: float64
```

El rango de presión arterial parece normal. No vemos valores raros.

chol

In [208…
```
df.chol.describe()
```

Out[208]:
```
count    303.000000
mean     246.264026
std       51.830751
min      126.000000
25%      211.000000
50%      240.000000
75%      274.500000
max      564.000000
Name: chol, dtype: float64
```

No se observan valores extraños.

fbs

In [210…
```
df.fbs.unique()
```

Out[210]:
```
array([1, 0], dtype=int64)
```

Valores esperados. Todo ok.

restecg

In [212…
```
df.restecg.unique()
```

Out[212]:
```
array([0, 1, 2], dtype=int64)
```

Valores esperados. Todo ok.

thalachh

In [214…
```
df.thalachh.describe()
```

Out[214]:
```
count    303.000000
mean     149.646865
std       22.905161
min       71.000000
25%      133.500000
50%      153.000000
75%      166.000000
max      202.000000
Name: thalachh, dtype: float64
```

El rango [71..202] parece normal.

exng

In [216…
```
df.exng.unique()
```

Out[216]:
```
array([0, 1], dtype=int64)
```

Valores esperados. Todo ok

caa

In [219…
```
df.caa.unique()
```

Out[219]:
```
array([0, 2, 1, 3, 4], dtype=int64)
```

Valores en rango. Todo correcto

output

In [222…
```
df.output.unique()
```

Out[222]:
```
array([1, 0], dtype=int64)
```

Valores en rango. Todo ok.

Conclusión: No hay ni nulos ni valores que representen pérdida de información. No se realizará ningún tratamiento sobre los datos

---

In [ ]:
```

```

4. Análisis de los datos.

5. Representación de los resultados

Se realizará durante toda la práctica

6. Resolución del problema

In [ ]:
```

```